DOCUMENT RESUME

ED 240 702 EA 016 466

AUTHOR Shostak, Robert, Ed.
TITLE Computers in Composition Instruction.
INSTITUTION International Council for Computers in Education, Eugene, Oreg.
PUB DATE 84
NOTE 89p.; Revised and expanded version of ED 226 709.
AVAILABLE FROM International Council for Computers in Education, 1787 Agate Street, University of Oregon, Eugene, OR 97403 ($6.00 prepaid; quantity discounts).
PUB TYPE Collected Works - Conference Proceedings (021) -- Reports - Descriptive (141)

EDRS PRICE MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS Autoinstructional Aids; *Computer Assisted Instruction; *Computer Programs; Elementary Secondary Education; *Microcomputers; Prewriting; Programed Tutoring; Programing; Student Writing Models; *Writing (Composition); Writing Exercises; *Writing Instruction; *Writing Processes

ABSTRACT

This volume consists of nine conference papers and journal articles concerned with microcomputer applications in the teaching of writing. After a general introduction entitled "Computer-Assisted Composition Instruction: The State of the Art," by Robert Shostak, four papers are devoted to how computers may help with the writing process. In "Computer-Assisted Prewriting Activities: Harmonics for Invention," Hugh Burns describes the computer as a tool for helping writers discover, arrange, and style ideas by means of interactive questioning strategies. A similar heuristic device for writing about literature is described by Helen Schwartz in "But What Do I Write--Literary Analysis Made Easier." In "Computers and the Composing Process: An Examination of Computer-Writer Interaction," Earl Woodruff identifies three ways a computer can help students write: as consultant, questioner, and collaborator. Then William Wresch, in "Writer's Help: A System Approach to Computer-Assisted Writing," employs computer programs to help students pick a subject, answer questions about it, and organize their ideas in a rough draft. The next three articles concern courseware design, beginning with an overview on "Design Considerations" by Alfred Bork, followed by articles on "Evaluating Textual Responses" by Joseph Lawlor and "Developing Interactive Language Experiences for the Computer" by Gloria Kuchinskas. The volume concludes with an essay on "Courseware Selection" by Ann Lothrop, followed by a list of computer-related journals and biographical notes on the authors. (TE)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED240702

Computers in Composition Instruction

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

X This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY

D. Moursund

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Edited by
Robert Shostak

ICCE Publications

EA 016 466

This booklet is published by the International Council for Computers in Education, a non-profit, tax-exempt professional organization. ICCE is dedicated to improving educational uses of computers and to helping both students and teachers become more computer literate. ICCE publishes *The Computing Teacher*, a journal for teachers and for teachers of teachers. It also publishes over ten booklets of interest to educators.

The booklet prices given below are for **prepaid** orders. On other orders a $2.50 processing charge will be added.

QUANTITY	PRICE (U.S. $)
1-4 copies	$6.00 each
5-9 copies	$5.40 each
10-29 copies	$4.80 each
30-49 copies	$4.20 each
50+ copies	$3.60 each

Please place your orders with:

INTERNATIONAL COUNCIL FOR
COMPUTERS IN EDUCATION
University of Oregon
1787 Agate St.
Eugene, Oregon 97403
(503) 686-4414, 686-4429

Copyright © ICCE 1984
(First Printing, January 1984)

Table of Contents

Introduction 2

Computer-Assisted Composition Instruction: The State of the Art
Robert Shostak 5

The Writing Process

Computer-Assisted Prewriting Activities: Harmonics for Invention
Hugh Burns 19

"But What Do I Write?"—Literary Analysis Made Easier
Helen Schwartz 27

Computers and the Composing Process: An Examination of Computer-Writer Interaction
Earl Woodruff 33

Writer's Helper: A System Approach to Computer-Assisted Writing
William Wresch 45

Courseware Design

Design Considerations
Alfred Bork 53

Evaluating Textual Responses
Joseph Lawlor 59

Developing Interactive Language Experiences for the Computer
Gloria Kuchinskas 65

Courseware Selection

Courseware Selection
Ann Lathrop 75

Biographical Notes 84

Introduction

Teaching students to write has always been and remains one of the most difficult tasks facing the English teacher. The litany of problems associated with this task is familiar to all English teachers who are serious about teaching writing: finding a topic, writing effective sentences, organizing the paper, revising and editing.

In the last several years, a number of English teachers and others interested in teaching writing have been exploring the use of the microcomputer as a tool to help deal with some of these problems. In April, 1982 SWRL Educational Research and Development in Los Alamitos, California called together a small group of these interested practitioners to discuss and share the work they were doing.

A number of papers were prepared especially for this meeting, published as *Conference Proceedings*, and distributed in limited quantities. These papers are being reproduced in this volume in an edited form along with selected articles published over the past two years in *The Computing Teacher*, the journal of the International Council for Computers in Education. The purpose here is to make this expanded collection available to a wider audience.

This volume does not purport to deal with all of the problems of teaching composition. Rather it attempts to reflect specific interests of individuals working in the field. Nevertheless an attempt has been made to organize the material to help the reader focus on specific concerns. The opening article, "Computer-Assisted Composition Instruction: The State of the Art," serves as a general introduction to the volume by attempting to review the promising practices in each of the important stages of the process of writing—prewriting, composing, rewriting and editing.

The group of four articles in the section called *The Writing Process* deals with how computers may be used to help teach writing. The second group of three articles in the section called *Courseware Design* focuses on designing computer programs to assist in the teaching of writing. The volume concludes with a general overview of software evaluation.

In article one of the section on *The Writing Process*, Hugh Burns tackles one of the most difficult problems for young writers—getting started. Burns sees the computer as a tool for stimulating the production of ideas—in his words, "to help writers discover, arrange and style ideas." He uses an interactive questioning strategy which imitates three different systems of inquiry—Aristotle's topics, Kenneth Burke's dramatistic pentad, and the tagmemic matrix developed by Richard Young, Alton Becker, and Kenneth Pike—to help students generate and store ideas for topics they choose themselves.

The author of the second article, Helen Schwartz, is also sympathetic to the novice writer's problem of getting started. Her focus is writing about literature, and she employs an approach similar to Burns'. The heuristic she has developed directs students, through questioning, to write about what they perceive is happening in a piece of literature; share these ideas with other students (electronically); receive feedback from these students; and use this feedback to refine their original thoughts.

In the next article, Earl Woodruff moves away from the generation of ideas (prewriting) to employ the computer as an aide in the composing process itself. He identifies three different ways in which the computer can assist the student in producing a piece of writing—computer as consultant; computer as questioner; and computer as collaborator. In the first instance the computer assists the students in using what is already known but what the student may have difficulty using because of what Woodruff labels "cognitive overload." In the second instance the computer is used to intervene in the composing process and lead the student through higher order questions to consider the expansion of thoughts already expressed. The last approach literally uses the computer as "co-author." The computer is programmed to produce the content and linguistic form of individual sentences, while the student has the responsibility for structure and style of the writing.

The last article in this section, by William Wresch, literally attempts to use the computer to assist the student in producing a very rough "rough draft" of a composition. Wresch uses the computer as a tutor to encourage students to pick a subject, answer some specific questions about their chosen subject, and select a rhetorical approach to help organize the statements they gen-

erated in the questioning routine. The computer then reproduces the students' work in what could be described as a first draft.

The articles in the *Courseware Design* section have been placed in an order roughly based on their level of specificity in dealing with the design of computer programs for classroom use. Alfred Bork, in "Design Considerations," discusses what he considers to be some of the bad practices in the present design of computer courseware. He emphasizes the importance of screen design, timing, content and media issues in the development of educationally sound products. He makes a plea for serious developmental efforts and notes the need for research to identify what pedagogical problems can be effectively handled by computer-assisted instruction.

The second article, by Joseph Lawlor, describes the work he and his colleagues have done in attempting to produce computer-based materials for the production of text. He details the nature of problems faced by the instructional courseware designer when attempting to develop sophisticated programs. Then Gloria Kuchinkas in the last article in this section attempts to establish some program development guidelines for classroom teachers interested in creating simple interactive language experiences for use in their own classrooms.

The last article in this volume has been included to help classroom teachers judiciously select commercially prepared materials which might be helpful in facilitating the development of writing skills in their own composition programs. The author, Ann Lathrop, also includes a very helpful list of resources for those interested in locating these materials.

Robert Shostak
Florida International University

Computer-Assisted Composition Instruction: The State of the Art

by
Robert Shostak
Florida International University

I believe that any state-of-the-art discussion related to computer-assisted composition instruction should consider first some of the general problems we face in teaching writing. We all know that the latest in the litany of things that Johnny (and Jane) can't do is write. We know this because *Time* and *Newsweek* have told us it is so. As professional teachers we have known what the pedagogical problems are for a long, long time.

Teachers of writing have always faced the problem of numbers. A writer needs an audience. Being the reading audience for between 100 and 180 English students in a typical secondary school is not my idea of an ideal situation for teaching writing. I grow tired of trying to fool my undergraduates and experienced teachers at in-service workshops by attempting to make this burden seem insignificant and then selling them on "cutesy" ideas for surmounting the task of reading a seemingly endless number of student papers.

The crowded curriculum is another serious problem that detracts from the teaching of writing. Elementary school teachers complain of the need to emphasize the basic skills of reading and arithmetic, while the teaching of writing competes, in the little time remaining, with science, social studies, art, music, health, career exploration and what have you. Secondary English teachers echo similar complaints. Most owe their allegiance to literature but must also find time for basic skills, film, TV, language study, and writing.

Lastly and reluctantly, I must question our teaching practices in general. In a recent article Anthony Petrosky cites compelling research that indicates that teachers tend

> to monopolize . . . classes . . . with talking and writing . . . while . . . students listen, regurgitate, and seldom engage in meaningful discussions or compositions that are free of hidden agendas and require the critical thinking skills of interpretation, judgment, analysis, synthesis, and documentation with evidence. (*English Education*, February, 1982)

I think Petrosky is right. We have fallen in with a system that emphasizes product over process. We go for the fill-in-the-blank or multiple-choice type of assessment. We engage students in exercises that are purported to teach writing but are actually easy-to-grade substitutes for the real thing.

We have other problems. In the introduction to *Research on Composing: Points of Departure* (National Council of Teachers of English, 1978), the editors, Charles Cooper and Lee Odell, challenge the basic assumption that teachers of writing have an adequate understanding of the term *composition*. They point to the work of Richard Meade and W. Geiger Ellis who discovered that paragraph development as taught in textbooks could not be found in a large number of actual pieces of writing they examined. Cooper and Odell also cite Richard Braddock's study which challenges certain conventional assumptions about the time-honored use of the topic sentence—a technique that recently has found its way down to the more prestigious suburban preschools. The editors also refer to Janet Emig's work, which poses a serious challenge to the admonition of every true believer that thou shalt not write before producing a complete outline.

My point here is that researchers who have systematically investigated the nature of the writing process and professional writers who have expressed their own thoughts about what happens when they write have not only caused us to reconsider our own views but have also provided us with new insights to understand the act of composing.

Research tells us that rather than teach writing as a set of discrete skills (which, when learned in some predetermined scope and sequence, will eventually become internalized to be applied to the task of writing) we need to teach writing as a process. We need to guide the developing writer carefully through the stage of prewriting, into the act of composing, and finally through the difficult period of rewriting and editing.

I mention these general problems at the outset as a word of caution. We cannot look at computer-assisted instruction as a panacea. Teachers are not going to become good writing instruc-

tors simply because they have a unique new technology available to them. They must first understand what it means to be a writer —to experience both the pain and the joy. They must understand the process, and they must be able to integrate the latest innovations in instructional technology into an already sound writing program.

It is probably much too early to talk about the state of the art in computer-assisted composition teaching. To me, state of the art suggests that the latest product, process, or technique has been preceded by at least one working model—itself the result of long-term planning, development, and field-testing. At this point in time, I believe that practitioners are just beginning to produce the first working models. Some may call them state of the art if they wish—I prefer to describe them as promising practices.

What then are these promising practices? For purposes of organization I would like to refer first to those kinds of programs that seem to best support the first stage of the composing process, prewriting, then move on to the writing stage itself, and finally explore those programs directed at the last stage, rewriting and editing.

PREWRITING

Story Maker

What appears to be an exceptionally promising piece of programming for teaching writing in the elementary school has been produced by Andee Rubin of Bolt Beranek and Newman, Inc., a Boston educational consulting firm. The program is called *Story Maker* and focuses on the prewriting stage of the writing process —invention. I like *Story Maker* because instead of emphasizing drill with letters, words or phrases, it encourages children to concentrate on a whole text. More specifically, children are guided to consider the logical flow of a narrative, the role of examples in an explanation, and the communication of characters' plans in a story.

The theoretical basis for this approach is found in cognitive psychology's investigation of children's writing and reading. Studies have shown that children, when faced with high-level cognitive tasks, tend to concentrate on lower-level processes such as decoding in reading, or spelling and handwriting in composing. *Story Maker* attempts to free young learners from attention to these details so that they may concentrate on higher-level cognitive processes.

Basically, the program engages students in a creative story-making exercise by allowing them to choose options from already

written story segments. After all the decisions have been made, the child has produced a complete story which s/he can read and share with other students. The program also allows hard copy to be printed for further activities planned by the teacher.

What I especially like about this piece of courseware is that the child maintains control of the creative process while the computer handles the "bookkeeping" details. Through simple commands, the computer presents structured options to the child, keeps track of the choices, and displays and prints the completed story when the activity is finished.

The child controls the direction the story will take through the choices s/he makes. Very early in the program the child begins to learn that making one choice rather than another will influence how the story will flow as well as how it will end. Because this program provides an early experience with manipulating language at a high cognitive level, it seems to have a great deal of promise for developing the kinds of skills one needs to become an effective writer.

Aristotle's Topics

Another example of courseware that reflects the process approach to the teaching of composition is that developed by Hugh Burns of the Air Force Academy English Department. Burns found that many of his freshman cadets were having a great deal of trouble at the prewriting stage. They did not seem to have the time or ability to generate the ideas necessary as a precursor to composing.

To help his students become more productive at this preliminary stage of the writing process, Burns developed a computer-assisted program that permits students to interact with the computer for the purpose of stimulating the formation of ideas. The program uses a set of questions based on Aristotle's enthymeme topics to engage the student in a Socratic-type dialogue. The dialogue guides students through an explanation of their subjects in order to uncover new ideas, facts, opinions or arguments that heretofore they were unable to recover from memory.

I see this as a promising practice because Burns' findings suggest that students can be encouraged to increase the number and sophistication of their ideas, and that dialogue can help students express ideas more clearly. Any computer-assisted instruction that is effective in improving students' skills of inquiry will be welcomed by teachers of composition.

COMPOSING

Word Processors

The composing stage of the writing process is beginning to benefit greatly from several unique computer applications, although, strictly speaking, these applications might not be defined as computer-assisted instruction. It is no longer a secret that computerized word processing has become an important writing aid for authors, researchers, teachers and students. The ease with which word-processing packages assist writers in creating and editing text, even to the point of checking syntax and spelling, can all but eliminate the need for typewriters, pens, pencils, and all the rest of the usual paraphernalia associated with the tedious task of editing.

Creative teachers of writing have found several very effective uses for word processing. Children at the elementary school level who find the production of handwritten text both difficult and painful are being taught to master the computer keyboard, making entry of text much simpler and freeing children to concentrate on the creative aspects of writing.

One of the most difficult tasks of a writing teacher is to convince students that revision is important. As a general rule, students believe that what flows from their pens the first time is exactly what they want to say and how they want to say it. Why tinker with perfection? Now teachers are reporting that students introduced to the power of word processing are beginning to take more kindly to revision. The ease with which words, sentences, and even entire paragraphs can be deleted, inserted, and shifted around has helped students gain a greater understanding of and appreciation for the value of rewriting.

Research has for a long time suggested that to improve writing, one needs to write. For several reasons, all of them well known to experienced classroom teachers, it is difficult for students to produce the number and variety of composition assignments necessary. Current reports suggest that the use of word processing in teaching writing is beginning to encourage students to compose more and longer texts. Although there is no clear-cut evidence yet that students are writing better, the motivation to write is high, and students' affective responses to writing assignments are positive.

Compupoem

Working in the second stage of the writing process, composing, but coming at it from another direction are two poets who have developed different approaches to teaching writing. Both techniques deserve serious consideration by teachers of writing.

The first one was developed by Stephen Marcus of the University of California, Santa Barbara. Marcus has produced a poetry-writing program called *Compupoem*, which engages students in all stages of the composing process. Unlike many poem-generating programs that simply produce a poem from a data bank of poetic phrases randomly displayed, *Compupoem* asks students to think about what they want to say, and it even provides the opportunity to see instant reproductions of their poems in different formats. This computer-assisted approach to writing poetry permits students to generate ideas or think about what they want to say first. In addition, students can control how the various parts of the poem will fit together in the finished product.

The Electric Poet

The second poetry program I would like to discuss was developed by Edmund Skellings, director of the International Institute for Creative Communication at Florida International University. It takes the form of a very sophisticated piece of software called *The Electric Poet.*

For several years Skellings has been working on a computer program that would display color-paired symbols, numbers and letters, using an ordinary color television screen. He was recently awarded a patent for the process, which he calls the Binemic System. What the system does is to permit color to be placed on a video screen to produce color pairs or groups illustrating relationships between letters or symbols.

Other features of the system include the capability of displaying letters in an infinite variety of patterns on the screen (similar to the patterns e.e. cummings uses on the printed page) and producing animation. Thus, a poet can easily use a particular pattern for displaying words while at the same time controlling when, where and how quickly these words will appear to the reader experiencing the poem. To add to the flexibility of the display, letters may be made to appear as if they are three-dimensional on the screen.

What Skellings has accomplished is a blissful marriage of technology and poetry to produce a truly unique literary experience. Perhaps a more appropriate term might be a "liter-media" experience. Skellings has released poetry from the limitations of the one-dimensional, black and white printed page. He has added a powerful new dimension to the creative process, increasing the poet's opportunity to appeal to multi-sensory experiences.

Skelling's computer poem, then, is one that is *viewed* rather than read or heard. It may be seen in various colors. It may appear on the screen in its entirety or take advantage of the time

dimension and be displayed word by word, letter by letter, or thought by thought. Finally, the poem may even appear to move as it is presented on the screen. Computer poetry, for me, exemplifies the new rapprochement of technology and the humanities that we are seeing today in numerous university programs around the country.

Skellings believes that his Binemic System also has many practical applications for educators, especially those who teach language arts. He bases this belief on studies done by researchers in brain function imbalance. It is pretty well accepted now that the two halves of the brain control different functions—the left hemisphere specializing in spoken language, reading and writing; the right hemisphere dealing with spatial relations and musical patterns. Skelling says that when both parts of the brain are fully activated and integrated with each other, optimal conditions for creativity exist.

Consequently, teaching methodology using the Binemic System can engage both brain hemispheres because it simultaneously displays text in a linear-sequential format and in a color pattern-recognition format. Such methodology produces optimal conditions for effective learning to take place.

Specific subject matter that Skellings would like to develop for field-testing his system includes figurative language, patterns of thought, poetry, grammar and reading. He thinks that students who understand various kinds of figurative language will be able to make comparisons more vividly. Such students will also have options for using more than literal language in describing and in understanding descriptions. Skellings suggests using his program to define and illustrate by color-coding various figures of speech.

Because critical thinking is so important to becoming a wise consumer and thoughtful voting citizen, Skellings also hopes that teachers will use his program to color rhetorical devices used in advertising, persuasive writing and political campaigns so that these devices may be identified, defined and studied to determine their effect.

Another effective use of this program could be illustrating, by means of color analysis, the various patterns of thought writers use to express their ideas, i.e., exposition, persuasion, description and narration. Many possibilities exist for developing color-coded writing models that could be visually taken apart and put together again instantly by the computer—a powerful tool for the teaching of writing.

Skellings has already demonstrated the viability of color-coding poems so that students can recognize the elements that lend meaning to poetry. This approach very effectively illustrates the relationship of language links and structure to meaning.

REWRITING AND EDITING

Next, let me describe for you what I think are some of the most promising practices for teaching the last stage of the writing process—rewriting and editing. Each program in this last group either deals directly with text that has already been created and needs to be revised or may be used to help students develop a higher level of sophistication with specific skills they have already demonstrated in their writing.

The Navy Program

The first program I would like to discuss deals with two important elements of writing—organization and development of style. Robert Wisher of the Navy Personnel Research and Development Center in San Diego has been working on both reading and writing programs over the past several years. One of his computer-assisted programs leads students through a procedure for organizing sentences into a meaningful paragraph. Students are presented with several sentences, each preceded by a number. First, students must identify which of the sentences they believe to be the topic sentence. Then, they rearrange the remaining sentences by selecting the numbers preceding each sentence. The computer instantly displays the organized paragraph. Next, students are allowed to edit their paragraphs in a variety of ways in order to clarify meaning and provide stylistic effects. This is accomplished by a series of prompts that allow students to position the topic sentence, place short sentences before longer ones, delete unimportant sentences, and insert a single sentence of their own.

Wisher also developed a program to assist students in developing better control over a variety of sentence constructions. The program is based on some of the sentence-combining research conducted over the past several years. Similar to the paragraph-organization programs, this piece of computer-assisted instruction has the student combine phrases into meaningful sentences. The student can see immediately how phrases can be combined to form clauses and how clauses can be combined to form sentences. There is even an option built into the program that allows the student to write a sentence with a particular style, for example, a sentence that begins with a specific verbal phrase or that contains both a dependent and independent clause. Both of these programs help a student to understand the organizational aspects of writing and appreciate the stylistic options open to the writer.

SWRL

Others working along similar lines are Bruce Cronnell, Ann Humes and Joseph Lawlor at SWRL Educational Research & Development. Their approach is also one that recognizes writing as a process. Their developmental efforts have been concentrated on four specific elements of writing instruction: sentence combining, generating content for a particular discourse type, organizing content and revising.

RSVP

Still another approach to assisting students in the rewriting and editing process is the very sophisticated *RSVP* (Response System with Variable Prescriptions) developed under the direction of Kamala Anandam at Miami-Dade Community College in Florida.

The best way to understand how *RSVP* is used is to assume that you have just collected 75 esays from three of your classes. First, you must read the essays holistically in order to place them in one of four levels A, B, C or D. These levels correspond to levels of writing skill: primary, basic, intermediate and advanced. The next step is to ready the essays analytically and identify those specific errors you wish to call to the attention of the student. At this point you may include any written comments you deem appropriate on individual papers.

Once the readings are completed, you indicate on a computer card the kind of feedback you wish the student to receive. The computer then generates a personal letter for each student, reflecting the items that you coded on the computer card.

The feedback each student receives is written on a reading level that corresponds to the student's writing skill level, which was determined by the teacher in step one, i.e., fifth grade (primary), seventh grade (basic), ninth grade (intermediate), or eleventh grade (advanced). The feedback consists of written prescriptions that reinforce for the student what has been accomplished and provide instruction in how to avoid errors in future assignments. *RSVP* is capable of providing feedback in a variety of skill areas, including spelling, subject-verb agreement, sentence structure and organization.

The system is also capable of generating individualized study/exercise assignments. These assignments are based on twelve major writing skill areas, which have been divided into over sixty subskills. Students are also given a selection of page references in specific texts to use as resources.

In addition to these instructional capabilities, *RSVP* also has management capabilities. The program accumulates students' errors from assignment to assignment, provides a status report to

teachers, and contains an option to provide students with progress reports at times designated by the teacher.

The Writer's Workbench

Finally, I would like to describe to you what is perhaps the most advanced editing system in existence today, *The Writer's Workbench*, a creation of Bell Laboratory scientists in New Jersey. After a writer enters a piece of text, the 32-program system is able to correct spelling, punctuation and grammar. It will also analyze style and provide feedback to the author on sentence length, cliches, wordiness and jargon.

The program literally suggests editorial changes to the writer. Based on traditional guides to writing effective prose, such as Strunk and White's *The Elements of Style*, the system searches a text for stylistic breaches and suggests alternatives. For example, the program will detect split infinitives or forced expressions such as "the utlization of" and offer the writer specific alternatives for improving the text.

Yet another unique feature of the *Workbench* is its prose comparison program. Stored in the computer's memory is a set of written passages that have been developed by the program's designers to reflect some specific standard of effectiveness for certain kinds of expository prose, for the most part, articles written by scientists. Some points of comparison used are readability level, average sentence length, sentence types, and passive verb phrases. After the computer compares an original text to one of its benchmark selections, it produces a count of each point of comparison along with a suggestion. For example, a writer might receive the following kind of report:

1. Readability (years of education necessary to read your text): 13.2
2. Average sentence length: 32.4 words
3. Sentence types: simple, 13 percent; complex, 47 percent. Your text contains more complex sentences than is usual for this kind of writing. You could improve your writing by reducing the number of complex sentences.
4. Passive verb phrases: 10 percent

Obviously, there are some real limitations to this approach to editing. The *Workbench* system reminds me of a brief program I wrote with my dissertation advisor many years ago. I was taking a criticism course at the time and was eager to impress my professor with the computer's ability to analyze prose style. In a few short hours I had written a very simple counting program to which we submitted two passages—one from *The Sound and The Fury*, by Faulkner, the other from *A Farewell to Arms*, by

Hemingway. The program was flawless in its ability to count words and sentence length. It was less than spectacular in the advice it gave to each author for writing more effective prose. Faulkner was asked to take lessons from Hemingway. And Hemingway, in turn, was asked to take lessons from Faulkner.

Whatever its shortcomings, though, *The Writer's Workbench* does represent the state of the art in providing editorial assistance for specific kinds of prose.

A SYSTEMS APPROACH

To date one of the most ambitious projects undertaken to teach students to write is *Quill*, a joint venture of the researchers at Bolt Bernanek and Newman, Inc. and the Network, Inc. *Quill* is a systems approach, microcomputer-based writing curriculum for the upper elementary grades, which attempts to organize learning around a set of interrelated programs that lead students through important steps in the writing process. The system consists of six different programs which focus on planning, integrating reading and writing, publishing, communicating electronically, writing with peers, and revising.

The first program, *Planner*, takes into account the centrality of planning in the writing process. The program helps students to take notes, write down ideas, and organize the thoughts they will later develop into a piece of writing. More specifically the *Planner* is used to create a set of questions students can use to stimulate their thinking about what they have read, what they have observed, or what they have viewed on film or some other medium. The teacher and students in a brainstorming session can create different kinds of formats to suit a particular activity and use the Planner to reproduce that format for later use by the students themselves when they need to go through the prewriting (planning) stage of a writing assignment.

The second program in the system is called *Library*. With this program the designers hope to integrate reading and writing by creating an information exchange. Based on the premise that writers frequently engage in research—by reading, this program permits students to create their own data bases. Not only do they create and enter the kinds of information they or their friends might need to access for a future writing assignment, but they also learn how to assign keywords to their data which others may use for easy information retrieval. The program's ability to enter, store, and retrieve information readily achieves the goal of integrating reading and writing in the same activity.

One of the serious problems student writers have is an attitudinal one. They feel that the only person who ever reads what they write is the teacher—and that really doesn't count. To

assure that students have an opportunity to gain a readership and to learn about writing with a purpose and for a specific audience, the *Quill* system uses the *Publisher*, a general utility program capable of creating formats for easy publication of newspapers, pamphlets and even books.

The fourth program in the system is *Electronic Mail*. Having the same pedagogical goals as the *Publisher*, this program adds a new dimension to the writing process—feedback. *Electronic Mail* not only creates an environment in which writers must consider audience and purpose, but it also anticipates a response. We know students have always enjoyed sending messages to each other— clandestinely. *Electronic Mail* now formalizes a natural language activity so that all students can benefit.

Story Maker is the fifth program in the *Quill* system and it has already been described in an earlier section of this article.

The last program incorporated in this package is a text editor. Called *Writer's Assistant* it is capable of correcting mechanical errors as well as moving text around or putting two pieces of text together to create a longer one. Another nice feature of this program is its ability to display all sentences in a student's text at the left margin. This allows the student to quickly scan for missing capitals, end punctuation, run-on sentences, sentence variety or fragments. With this kind of display students can easily discover their mechanical, grammatical or stylistic faults and with a few simple key strokes correct them.

The designers of *Quill* have already conducted a pilot test of their completed programs and field-tested the system during the 1982-83 school year. Additional field testing is taking place during the 1983-84 school year.

CONCLUSION

Computer-assisted composition instruction is still very new to most teachers of writing. The success of any new innovation in education depends on how well it is accepted by classroom teachers. I think that acceptance will largely depend on three things: (1) how creative we are in solving the problems related to numbers of students and time allotted for writing in the curriculum; (2) how successful teacher educators are in training teachers of writing to approach writing as a process; and (3) how convincing program developers are in demonstrating that computer-assisted instruction is, indeed, the state of the art in the teaching of writing.

PROGRAM INFORMATION

Story Maker

For an order form write to: *Story Maker*, Cheryl Liebling, Bolt Beranek and Newman, Inc., 50 Moulton St., Cambridge, MA 02238.

Target grades: 3-6

Hardware description: Apple II or Apple II+. Requires 48K, Applesoft ROM, and disk drive (2 are preferable).

Cost: $30

Compupoem

Stephen Marcus, The Southwest Writing Project, University of California at Santa Barbara, Santa Barbara, CA 93106.

Target grades: 4-college

Hardware description: Apple II+ or Apple IIe with disk drive.

Cost: $16.95 plus $1.00 for shipping. California residents add 6% tax. Prepayment preferred. No purchase orders.

The TRS-80 version in English and Spanish will be available winter 1983.

The Electric Poet

U.S. Great Britain and Canadian patents on the Binemic System cover the display on any CRT, the process of creating the display on the screen and the teaching method. Software for the Binemic System which will include *The Electric Poet* is being developed by Control Color Corporation, 700 E. Dania Beach Blvd., Dania, Florida 33004. Complete packages will be available in 1984.

RSVP (Response System with Variable Prescription)

Hardware description: IBM 370

Cost: $3000

Camelot: The Individualized Information System

(Microcomputer version of *RSVP*)

Hardware description: IBM PC with 10/20 megabyte hard disk, QUAY 900 with two quad density 8" disks, ALTOS 8000 with 10 megabyte hand disk, TRS-80 Model II with 3 double density 8" disks.

Cost: Contact Cindy Eliot for pricing information.

Auxilliary Services, Miami-Dade Community College, 11011 S.W. 104th St., Miami, FL 33176.

For information on program development write to Dr. Kamala Anandam, Division of Computer-Based Instructional Development and Research, Miami-Dade Community College, Miami, FL 33176.

Target age: High school to college.

UNIX* Writer's Workbench**

Western Electric Company, Inc., Software Sales and Marketing, P.O. Box 25000, Greensboro, NC 27420. (Available to qualified institutions under an Educational Software Agreement on a per CPU basis.)

Hardware description: Requires UNIX operating system.

Cost: Initial CPU—$2,000; additional CPU—$800.

Quill: Microcomputer-Based Writing Activities

For information on the results of the pilot tests and the availability of *Quill*, you may write to Bolt Beranek and Newman, Inc., 50 Moulton St., Cambridge, MA 02238.

*_UNIX_ is a trademark of Bell Laboratories.

**_Writer's Workbench_ is a trademark of Western Electric Company.

The Writing Process

Computer-Assisted Prewriting Activities: Harmonics for Invention

by
Hugh Burns
United States Air Force Academy

I expect to be teaching composition until at least 2011. Therefore, I expect to hear the question, "What should I write about?" a few more times. Specifically, at 140 students for twenty-nine years at eight essays apiece—that's 32,480 times. Think of the challenge I'll have coming up with 32,480 different topics. I'm not prepared for that, but I do think the analytical engine that has crept into our humanities garden can help me. In fact, I cannot imagine my 9:30, Tuesday composition course in the spring of 2006 without computer terminals somewhere, maybe everywhere. I'll use computers to train teachers. I'll consult and advise student writers and some professional writers on disks. I'll accept software as I accept research papers today. Yes, the electronic revolution is here.

Now the most vital part of my work and my daydreaming is in envisioning and promoting the computer as a writer's tool, a tool to help writers discover, arrange and style ideas. The research I've been most involved in thus far has to do with stimulating rhetorical invention in composition using computer programs

that ask questions and motivate a systematic inquiry. Not only have such programs modified my own writing, but I'm learning to think about the geometry of invention in new ways. Some writers invent in solos, others in duets, and some in orchestration.

INVENTION AS SOLO

Our students have voices. They have experience. What many of them don't have is the strategy or the confidence to amplify the voice or articulate the experience, so learning to inquire into their own experience challenges them. That elusive search for self-confidence short of wisdom often is the most difficult step in the writing process. We listen to our students' solos, wishing they had more time to practice. Invention is a solo—first a solo based on knowledge and experience.

But this solo uses invention techniques: some we can describe and some we cannot. Those we can describe are often called heuristics, heuristics being strategies for discovering ideas or solutions. If the imagination is dark, an invention strategy is a way of flooding the stage with light and filling the auditorium with sound. The creative enterprise of stimulating the process of inquiry will challenge the student, whatever the curriculum—anthropology or computer science, English literature or organic chemistry. I don't envision a classroom without teachers, but I will not, cannot, must not "teach" the discoveries of prewriting. The prewriter is wrapped up in the undiscovered self, and the responsibility for solving the invention puzzle is ultimately each student's. *But* others can stimulate the process; others can make invention more of a duet.

INVENTION AS DUET

Needless to say, the life of the mind, too, can be a duet of language and thoughts, thoughts and language. However, the duet I'm speaking of is the duet with teacher and student. A teacher can direct students' attention and experience to the topic in a number of ways. And humanistic teachers can do that without knowing what students know about a topic, without caring as much as students care about a topic, and without actually partaking in the discoveries students will make about a topic.

Say I have a student who is the world's chief authority on indolebutyric acid, a plant hormone especially good for root development. As a teacher I can ask him or her questions that will help stimulate thinking about that acid. For example:

Who discovered the acid? When?

Was the process in any way accidental?

How does that acid change?

What special experience made you select indolebutyric acid as your topic?

How is that acid created?

What systems benefit most from this acid?

What implications does this acid have politically? Economically? Culturally?

What is the opposite of this acid?

How is this acid like an accomplice?

Like a ruby-throated hummingbird?

Like a hearth?

What will you claim at the end of your essay about indolebutyric acid?

The suggestive power of such questions doesn't depend on the content or the topic but rather on the interrogative propositions themselves. The student must have the content. The student must be the person who has prepared, who has incubated, who has a felt need for the insight, and whose own curiosity will finally evaluate the quality of the findings.

Consequently, a teacher may ask these questions without fear of pretending to know the answers (though some of us probably couldn't resist the temptation to explore some of the more intriguing issues, say the analogy to the hearth, suggesting that the temperature variations caused by the acid could possibly change the developmental cycle of a root system for the better). Inquiry tempts us; inquiry should tempt us. Questions tempt most of us to answer; that, as a matter of fact, is a question's propositional value.

The difficulty with invention as duet is that a composition teacher's voice cannot, practically speaking, last that long. Our enrollments are too high; our time is not our own. It would be impossible to talk to all of our student writers in order to spur their creative processes. Consequently, if we wish to engage them in the prewriting process, we need help. That's where the computer entered for me. In January, 1977, in the midst of my doctoral research, I began to think about rhetorical invention as an electronic orchestra.

INVENTION AS ELECTRONIC ORCHESTRATION

I have designed, developed and programmed three computer programs to encourage thinking, programs I call "artificial intuition." These programs ask writers questions based on particular

systems of inquiry, specific heuristics—though at some point I hope students will say to themselves, "Wait a minute, I can ask myself such questions." The programs imitate three popular heuristics, specifically Aristotle's topics, Kenneth Burke's dramatistic pentad, and the tagmemic matrix developed by Richard Young, Alton Becker and Kenneth Pike.

The strength of these programs is that they imitate a way to investigate a topic, not *the* way by any stretch of the imagination, just one way. Prewriting, moreover, has certain subroutines that a computer program can imitate rather well. What are they?

1. A program can ask the question.
2. A program can clarify the question.
3. Good software can define the dimensions of the question.
4. The software can call attention to the essay's purpose.
5. It can purposefully distract (for incubation's sake).
6. It can rephrase the question.
7. It can create random metaphors.
8. Invention programs can offer research suggestions.
9. A program can print a copy of the dialogue so a student can later evaluate the answers.

After a prewriter uses such routines, then the teacher can jump in more efficiently as an inquiry diagnostician and tell students what cognitive styles they have already used, suggesting others that may help them develop more ideas. Such conversations then become the motivation for furthering the inquiry.

Now, you need to see an example. In the following computer-prompted dialogue between a senior enrolled in our literature seminar and the Aristotle TOPOI program, Peggy explores Thomas Pynchon's *The Crying of Lot 49* and his symbolic use of the entropy concept. We join her after a few introductory remarks—directions and such. Peggy's remarks are in bold type.

NOW I NEED TO FIND OUT WHAT YOU ARE WRITING ABOUT. SO WOULD YOU PLEASE TYPE IN YOUR SUBJECT. I AM LOOKING FOR ONE TO THREE WORDS.

ENTROPY IN MAXWELL'S DEMON

HOLY ELECTRONICS! THAT'S WEIRD, I USED TO DATE A COMPUTER INTERESTED IN ENTROPY IN MAXWELL'S DEMON.

A COMMENT ABOUT PURPOSE

DURING THIS EXPLORATION PROCESS, YOU WILL BE ASKED TO CLARIFY THE PURPOSE OF YOUR PAPER ON ENTROPY IN MAXWELL'S DEMON.

SO NOW WOULD YOU BRIEFLY DESCRIBE WHAT THE PURPOSE OF YOUR PAPER IS BY COMPLETING THIS STATEMENT: THE PURPOSE OF THIS PAPER IS . . . (LIMIT: ONE LINE)

DESCRIBE HOW THE ENERGY THAT IS PUT INTO THE "DEMON" VIA THE "SENSITIVE" IS TRANSLATED.

ANY MORE?
IF SO, TYPE WHATEVER IT IS. IF NOT, TYPE NO.

AND THEN LATER PUT TO USE AS ENERGY OUTSIDE THE CLOSED SYSTEM.

FINE, PEGGY, YOU AND I WILL TALK AGAIN ABOUT YOUR PURPOSE.
RELAX NOW, PEGGY, AND ENJOY THIS BRAINSTORMING SESSION.
WHAT HAS BEEN DECIDED ABOUT ENTROPY IN MAXWELL'S DEMON TO DATE?

EXPLAIN.~

DECISIONS HAVE BEEN MADE ABOUT ENTROPY IN MAXWELL'S DEMON.
WHAT WERE THEY ABOUT? WHO MADE THEM?
FOR EXAMPLE, IF I WERE WRITING A PAPER ABOUT INFLATION, I WOULD WANT TO WRITE A PARAGRAPH OR TWO ABOUT THE GOVERNMENT'S LEGISLATION TO DATE.
TRY ANSWERING THIS QUESTION NOW.

ENERGY IS PUT INTO THE "DEMON" VIA THE "SENSITIVE." NEXT THE DEMON WOULD TRANSLATE THE "COMMUNICATION" AND SEPARATE IT INTO DIFFERENT ENERGY FORMS.

GOOD, PEGGY, ADD TO YOUR RESPONSE NOW.

CONTINUE!

(IF YOU DON'T UNDERSTAND, JUST SAY SO NEXT TIME. I'LL HELP.)
WE'RE MOVING RIGHT ALONG. HERE IS QUESTION 2.
WHAT ARE THE GOOD CONSEQUENCES OF ENTROPY IN MAXWELL'S DEMON?

IT CAN CREATE ENERGY IN A CLOSED SYSTEM (WHICH DEFIES THE 2ND LAW OF &&

GO ON, PEGGY.

THERMODYNAMICS).

FINE, PEGGY. WRITE SOME MORE.

THIS "DEMON" CAN BE COMPARED (ON A VERY BROAD LEVEL) TO THE DEVIL IN . . . THE WORLD TODAY. THE DEVIL TAKES THE ENERGY COMMUNICATED TO IT . .. (BY PORNOGRAPHY, OR SIN FOR EXAMPLE AND USES IT TO

CREATE (CHANNEL) IT TO OTHER USES—TO WIDEN HIS CONTROL OF THE WORLD.

OUTSTANDING, PEGGY.
ANYTHING ELSE?
(YOU CAN ADD MORE INFO, ASK A QUESTION, OR GIVE A COMMAND—WHATEVER YOU WISH.)

CONTINUE!

OKAY.
DO YOU WISH TO NARROW OR CHANGE YOUR SUBJECT?
(MAYBE REVISE THE WAY IT SOUNDS IN THESE QUESTIONS?)

NO.

• • •

WHAT OBJECTS DO YOU ASSOCIATE WITH ENTROPY IN MAXWELL'S DEMON? HOW MIGHT THEY BE INCLUDED IN YOUR THEME?

EXPLAIN!

IF I SAY "BLACK," YOU SAY "WHITE."
IF I SAY "HEADACHE," YOU SAY "ASPIRIN."
NOW, PEGGY, IF I SAY ENTROPY IN MAXWELL'S DEMON. WHAT DO YOU SAY?
WHAT ARE YOU THINKING NOW?

THE IMAGE OF THE PERPETUAL MOTION MACHINE COMES TO MIND—MAKING SOMETHING OUT OF NOTHING.

BY GEORGE, PEGGY, GOOD ONE. WRITE A LITTLE MORE PLEASE.

ARRRRGGGGG

FANTASTIC. ANYTHING ELSE?

HELP!

OKAY.
WHAT IS THE MOST LIKELY PLACE FOR ENTROPY IN MAXWELL'S DEMON TO EXIST?

WHAT DO YOU MEAN?

WHERE SHOULD I GO TO SEE ENTROPY IN MAXWELL'S DEMON? CAN I GO INSIDE? CAN I GO OUTSIDE? WHY OR WHY NOT?
YOUR TURN, PEGGY.

OH MAN

AHHHH, SHORT AND SWEET. NOW TELL ME WHY?
IN OTHER WORDS, ELABORATE A LITTLE.

I GUESS THAT THIS REALLY DOES BROADEN THE WHOLE QUESTION. THE WHOLE . . . BASIS FOR THE ABILITY FOR THE DEMON TO WORK WOULD LIE IN THE PERSON . . . WHO WOULD BE TRYING TO "COMMUNICATE" TO THE DEMON. THIS WOULD . . . THEREFORE MEAN THAT WHOEVER WAS THE "SENSITIVE" WOULD ALREADY, TO AN EXTENT, BE UNDER THE DEVIL'S CONTROL. NOW I'M REALLY CONFUSED.

. . .

STOP!

YOU EXPLORED 4 QUESTIONS OUT OF THE 11 I ASKED. THAT'S 36.3636 PERCENT. LET ME REMIND YOU THAT YOU ARE STILL IN THE FIRST STAGES OF THE CREATIVE PROCESS. THESE IDEAS MUST SIMMER NOW. ALSO, I HOPE YOU CAN CREATE SOME OF YOUR OWN "TOPIC" QUESTIONS. I WON'T ALWAYS BE AROUND TO HELP.
HOPE YOUR PAPER IS TERRIFIC!
GOOD BYE & GOOD LUCK!

Now I'll grant that Peggy's answers reflect a certain caring about her subject. She wasn't putting in "garbage," and she mentioned afterwards that she wished she had known more about the topic before she started the inquiry. Nevertheless, her computer-assisted dialogue lasted about forty minutes, and when she left the lab, she took twenty-one pages of printout. She had responded, and although she may not have fully comprehended it, she was now incubating—the classic second stage of the creative process. The orchestra was tuning up.

SOME FINAL HARMONICS

As for the research findings, I'll only mention two items.* First, one of the major dilemmas: the problem of "rhetorical overload." In the experiments using these programs, students had more ideas about their topics, for all of the quantitive gains were significant over a control group. But when these same students were asked to organize their ideas into some "arrangement," their arrangements were not significantly better than the control group's arrangements. Arrangement, for these students, did not develop naturally from lots of ideas. In fact, too many ideas made the arranging task more difficult and more time-consuming. Second, we also verified that invention doesn't end. Several students told us on the follow-up questionnaire that the *best* answers to the questions did not occur to them while they were on-line. They were stimulated to think about their subjects from new points of view. That's exciting.

The future of computer-assisted instruction, for me, depends on how well we can "open" the programs. The computer can be

used creatively and suggestively in the invention process. And our work continues. These days, I'm designing composition programs using graphic analogies, for right- and left-brain connections are worth investigating. Also, I'm trying to define the explicit link between invention and arrangement; I'm not as clear or as naive about that relationship as I once was. And what we'll be able to do with word processors in a composition setting will soon overwhelm us. As for the rest of the time? I daydream about the future.

So the year 2011 arrived. Somehow, I didn't think it would ever come. So here I am reading my last college theme. I'm so glad my students haven't asked me what to write about since 1997; if anything, they have learned to "look" inside themselves for their topics, to deal with their topics as curious inquirers. Look at this topic. Who would have ever thought that some sophomore would write an essay on "Indolebutyric Acid as Entropy" of all things. I wonder where she came up with that idea?

PROGRAM INFORMATION

Aristotle TOTOI

Listings and information on the development of all three programs are contained in Major Burns' dissertation, "Stimulating Rhetorical Invention in English Composition through Computer Assisted Instruction." University Microfilms International, 300 N. Zeeb Rd., Ann Arbor, MI 48106.

*A summary of the dissertation research can be found in "Stimulating Invention in English Composition Through Computer-Assisted Instruction," *Educational Technology*, August, 1980, pp. 5-10.

"But What do I Write?"— Literary Analysis Made Easier

by
Helen J. Schwartz
Oakland University

When I teach an introductory literature course at Oakland University and announce the usual written assignments, I sympathize with students when they ask, "But what do I write?" Even when I explain in class and pass out sample papers, students sometimes have trouble getting started. Although there's nothing like individual tutoring and feedback for helping such students, many students (especially at a large commuter school) find it difficult to schedule a conference time with me or with peers that could get them going on a paper. Other students find a class discussion or individual conference too intimidating.

For such students I've developed a CAI program called SEEN —the Seeing-Eye Elephant Network. The program helps a student develop a character analysis for literary studies. The Seeing-Eye section of the program is a tutorial which helps a student "perceive" what s/he has "seen"; here students are required to write, in response to prompts from the computer, a "notice" which is then posted on the "Network." The Network section of the program allows students to see and write "comments" on the "notices" posted on the "Network" by other students. The Elephant is the section of the program which records and remembers what the student has written (or revised) in a "notice" or in a "comment." These "notices" and "comments" are stored as textfiles which can be loaded onto an Apple-compatible word processing system for printing or revision into an essay.

The program is based on the principle that learning should be *active*—the learning should take place in the *student*, not the programmer. Therefore, the program does not judge the responses of the student—either in the tutorial or on the network. The aims of the program are:

1. To provide students with thought-provoking prompts so that they can generate ideas,
2. To encourage the refining and expansion of their ideas by giving and getting feedback (anonymity is provided through the use of pen-names),
3. To encourage students to learn *how* they have learned by providing a record that traces the development of their ideas.

Let's see how one student actually used the program in her world literature class. During a time scheduled for her convenience, she responded to the program's prompts, hypothesizing that Satan in *Paradise Lost* is "tricky." The tutorial then asked Kami (her pen-name) to provide various kinds of evidence to support her hypothesis. (The computer's questions are in capital letters and Kami's answers follow in bold type.)

WHAT DOES SATAN DO THAT SHOWS SATAN IS TRICKY?

1. He enters the body of a serpent to disguise himself.

2. He convinces Eve to eat from the tree of wisdom by telling her lies about it.

3. He is able to convince Eve that he is a wise serpent and not her enemy.

The program then asked Kami for further evidence that Satan is tricky—because of 1) what he says, 2) how others react to him, 3) how his behavior compares to others in a similar situattion, and 4) what the third-person narrator (if any) says to him.

Users get numerous chances to modify and edit a "notice." The program does not judge spelling or ideas, but it does, at this point, encourage self-criticism by asking students to look at evidence critically. For example:

OK, NOW TRY TO THINK OF ANY EXCEPTIONS OR CONTRADICTORY EXAMPLES.

WHAT EVIDENCE SHOWS THAT SATAN IS NOT TRICKY?

Adam and Eve knew that there is someone out to get them.

AT THIS POINT DO YOU WANT TO
1 = CHANGE LIST ENTRY (& ERASE EXCEPTION JUST LISTED)
2. EXPLAIN THE APPARENT CONTRADICTION

3. ADD THE EXCEPTION WITHOUT COMMENT

DO YOU WANT 1, 2 OR 3?

2 [carriage return]

COMPLETE THE FOLLOWING STATEMENT:

ADAM AND EVE KNOW THAT THERE IS SOMEONE OUT TO GET THEM BUT . . .

They don't know what he will do or what he will look like.

When Kami completes this part of the tutorial, she has already generated enough ideas to begin an essay on Satan.

As the example demonstrates, this CAI program lends itself to various analytical subjects. What follows outlines the general format of the analytical approach, the application for character analysis and selections from Kami's actual responses:

General format	**Particular application with sample response in parentheses**
[Pick an X]	Name a fictional character X in a literary work (X=Satan in *Paradise Lost*)
[Create hypothesis: X=Y]	Describe the character X by completing X is _____Y_____. (Satan is tricky.)
[Argue that X=Y]	Provide evidence to show that X is Y:
[Regarding type A evidence]	What does Satan *do* that shows Satan is tricky? (He enters the body of a serpent to disguise himself.)
[Regarding type B evidence]	What does Satan *say* that shows Satan is tricky? (He tells Eve lies about the tree of wisdom to get her to eat from it. . . .)
[Regarding type C evidence]	How do *others* react to Satan that shows Satan is tricky? (God had originally thrown Satan out of heaven because he didn't trust him.)
[Regarding type D evidence]	How does Satan *compare* to others in a similar situation and how does this show Satan is tricky? (The other angels do not compare themselves to God and are not thrown out.)*

*Note that the program does not (and cannot) respond to inaccurate or unclear responses. That is a job for *people* who respond on the Network.

[Regarding type E evidence]	If there is a third person narrator, what does he say that shows Satan is tricky? (He says that Satan chooses the subtlest of beasts for his embodiment.)
[Consider conflicting evidence]	What evidence shows that Satan is *not* tricky? (Adam and Eve know that there is someone out to get them.) In light of this evidence, you can now revise your notice, explain the apparent contradiction, or leave the exception without comment. [The student "explained" her contradiction by adding the following: (but they don't know what he will do or what he will look like.)]

The following questions have been added to SEEN after analyzing results of the field testing in which Kami took part.

[Does X change?]	Does the character *change* as time passes in the literary work, and if so, how?
[Summarize.]	How would you summarize your view of the character?

After Kami left the tutorial, her "notice" was recorded in her individual computer file, posted on the Network, and signed with her chosen pen-name. This qualified Kami to 1) see any "notice" on the Network, 2) add a "comment" (signed automatically with her pen-name) to any "notice," or 3) see any "comment."

On this particular occasion, after Kami posted her "notice," she got *indirect* feedback by looking at a "notice" with a similar thesis. In addition she opted to see a "comment" on this "notice" written by the instructor. The instructor's "comment" gave direction not only to the notice-writer, but also to other readers of the "comment" like Kami. Furthermore, Kami got support for her ideas from the "notice" she read because the thesis and evidence were similar to hers. She also recognized a difference, and this pushed her to analyze and add her own "comment" on the "notice" she read.

When Kami logged on again, she got *direct* feedback from two other students who wrote "comments" on *her* "notice":

HELEN: SATAN TELLS EVE THAT EVE WILL BECOME WISE AS HE HAS FROM EATING THE FRUIT. HE TELLS EVE THAT GOD IS AFRAID OF HER BECOMING LIKE HIM.

ZAPION: TRICKY SEEMS TO BE KIND OF A SOFT WORD FOR SATAN. HOWEVER, IT'S ABSOLUTELY TRUE.

Here Helen pushes Kami's argument along by providing more concrete evidence, and Zapion supports Kami but makes a criticism about her thesis, too.

Kami went on to write other notices, but her ideas about Satan, along with the comments that others wrote to her, were printed out for her use before the next essay exam.

The Seeing-Eye Elephant prompts students to verbalize what they've noticed in their reading. Then the Network helps them refine their ideas. Seeing other notices and comments shows that not everyone has the same reaction to a character in literature. This helps students understand what they need to explain and support in an essay. Also, writing comments on other students' ideas helps sharpen critical ability. And reading comments on their own work can help students refine ideas.

Notices and comments show that students, *especially* those having trouble on exams, improve their work. And the students enthusiastically commented on the value of the program in helping them develop their ideas and internalize a new and more critical way of learning:

> "I find myself looking for facts on characters as I'm reading the story through even the first time. I understand the content of the story better as I find these facts."
>
> A shy student writes: "I enjoy the computer—it lets you express your feelings about a character when you're by yourself and you have time to think about it and give your true opinion; there is not group pressure to answer questions."

This interactive program lets students concentrate on their ideas while showing them and leading them through a process tailored to the approach of the discipline. SEEN helps with the questions and the clerical work, but the *students* provide the ideas, respond to them, and revise them.

PROGRAM INFORMATION

SEEN
Helen Schwartz, P.O. Box 911, Rochester, MI 48063.
Target Audience: Senior high school through adult.
Hardware: Apple II+ (DOS 3.3, 48K) with monitor, one or two disk drives and a printer (with a word processing program optional).
Price: Write Helen Schwartz for purchase agreement and cost.

Computers and the Composing Process: An Examination of Computer-Writer Interaction

by
Earl Woodruff*
The Ontario Institute for Studies in Education

*With Marlene Scardamalia (York University) and Carl Bereiter (The Ontario Institute for Studies in Education).

Suppose you are asked to help a school child write a short composition. There are a number of different helping roles you could assume. One approach would be to act as a consultant, sitting quietly and only giving information or opinions upon request. At the other extreme you could actively suggest content and language, leaving the child with little more to do than select from your offerings. Between these extremes you could avoid direct suggestions of what to write but try to help the child by asking questions. However, there is a range of possible questioning roles —from asking very leading questions of the "Don't you think . . .?" variety to asking more open questions that leave most of the thinking to the child. Deciding on what kind of helping role to assume depends on at least two considerations: First, what is your objective—to help the child produce a good composition in the short run or to help the child learn? Second, which of the possible roles does your own knowledge and ability equip you to handle? Although on principle you might favor the role of asking open questions, it might prove difficult to ask educationally productive questions unless you had an unusually deep understanding of the composing process.

In designing ways for a computer to help children write, we have to ask the same two questions in order to determine a reasonable and productive role for the computer. As to objectives, most current uses of computers as word processors are aimed at helping the child produce a good composition in the immediate instance and only indirectly with helping the student learn. The programs I shall discuss, on the other hand, were aimed at interacting with students in ways to produce learning and only secondarily in ways that might lead immediately to a better composition.

The possible roles for a computer are, however, severely limited by the computer's capabilities and by the extent of knowledge of the composing process that can be applied to program design. The first program that we of the York/OISE Writing Research Project produced was one that took advantage of a consultative format by offering a "menu" of help selections. The second program we wrote used a questioning format. This program tried to lead the students to consider relevant questions as the composition developed. Finally, the third program we wrote assumed a collaborative role. The student had a set of responsibilities and so did the computer. Each of the three programs attempted to boost the students to a more analytic and comprehensive view of their texts.

THE COMPUTER AS CONSULTANT: CAC-1

Our initial program, CAC-1 (Computer-Assisted Composition), tried to assist students, not by taking over parts of the task, nor by specifying what to do and when to do it, but rather by helping students use knowledge they had, but were failing to use because of cognitive overload. This form of assistance, called procedural facilitation, grew out of our investigations of the cognitive processes underlying composing. Such facilitation has been shown to be an effective means of assisting the composing process (see Bereiter and Scardamalia, 1982).

CAC-1 was an attempt to reduce the mental burden that students are under while composing by integrating various types of help with a text collector and a storage routine. The program displayed a menu of available types of help whenever the "help" key was pressed. One of the types of help, for example, was a delineation of the genre elements a student could include. The program was tailored specifically to the opinion essay genre, and so the elements were things like statement of belief, reason and example. Having selected one of these, the student would then receive further help in the form of an explanation of the element and suggested sentence openers such as "One reason is. . . ." or "Some people think. . . ." to introduce the element.

Specifically, CAC-1 acted as a text editor until the "help" key was pressed, and when it was, the program would ask, "May I help you?" If the student answered *yes*, CAC-1 could assist in the ways described below.

1. **Following an argument plan.** If this form of help was requested, the computer would indicate that the following list of elements could be included: a statement of belief, explanation of the belief, reasons for the belief, opposing beliefs and refutations, more reasons and examples for one's belief, and finally one or two summary statements. When the student indicated the element s/he would like to include, the program presented an elaborated explanation of the selected element with one or two sentence openers appended (see Figure 1 for an example of this elaboration).
2. **Help in producing the next sentence.** This algorithm would search the student's last sentence for selected keywords and, upon finding one, would prompt the student to say more about it. For example if the student had used the word "believe" in the sentence before s/he asked for this help, the computer would respond, "Let's say more about your belief so the reader will understand."
3. **Help in changing words.** When students asked for this help, the program would print out their texts while numbering each of the words. Once the point of revision was identified by number, the students could enter or delete the appropriate word, or a section of text could be inserted.
4. **Help in checking unsure words.** This routine lowered the constraints of text production by using the "#" key to flag words the student was unsure of as to spelling or meaning. Whenever a student was in doubt about the spelling, meaning or appropriateness of a particular word, s/he was instructed to type in a guess and end the word by typing the number-sign key. This routine then delayed the correction of such words until the text was completely entered, or until the student chose to correct them.

Figure 1.

An Elaboration, by CAC-1, of a Structural Element as Shown to the Student

REASON FOR BELIEF

In this section of the paper you tell the reader *why* you believe what you say.

These sentences usually have the word *because* in them.

After you tell why you believe what you say, it is always a good idea to *give an example.*

You can start your sentence with:

One reason _____ _____
or,
I believe this because _____ _____ _____

PRESS C TO CONTINUE

Experimentation with the above program on grade six students revealed mixed results (Woodruff, Bereiter & Scardamalia, 1981-82). While the students used, enjoyed and praised the program's beneficial effects, the overall quality of their papers was not improved. An apparent weakness of CAC-1 is that it is too easily assimilated to a low-level "What next?" strategy of composition (Bereiter & Scardamalia, 1982). This strategy is an overly simple, forward-looking approach to writing that fails to give attention to the whole text. Consequently, while the children found the program helpful in triggering ideas for what to say next, it did not seem to have engaged students in a higher-level consideration of composition choices. Although more experience and improved typing skills would likely have increased the program's beneficial effects, so might a program that takes a more active role in directing the students' attention between the higher and lower levels of the composing process.

THE COMPUTER AS QUESTIONER: CAC-2

Our second program, CAC-2, attempted to intervene more directly in the students' composing process with the intention of leading students away from their characteristic "What next?" strategy. Instead of waiting for the student to call for help, CAC-2 intervened at the end of every sentence. In point of fact students using the earlier program, CAC-1, typically asked for help at the end of each sentence. CAC-2 intervened without being asked. It guided the student through a question-and-answer sequence of variable length. The intent was to block use of the "What next?" strategy by getting the student to consider higher-level questions having to do with type of content, warrant for assertions, and possible reader reactions.

CAC-2 intervened with a question every time a sentence terminator was detected. The program contained 26 questions arranged in a branching sequence that depended on the answer to a preceding question. For instance, if the student responded *yes* to the question, "Do you have a reason for your opinion?" the computer would respond, "Okay, let's tell that to the reader." Upon detecting the next sentence terminator, the program would then display the question, "Have you made your reason clear to the

reader?" The student could answer each question *yes* or *no*, or could press a key instructing the program to move on to the next question or to hold the question until another sentence was completed. The overall intent of the questioning procedure was to lead children to compose more carefully considered and more fully developed opinion essays by switching their attention back and forth between the high and low levels of the process. In short the program tried to induce the student to consider many of the issues that mature writers consider but that are not taken into account by users of the "What next?" strategy.

An example of the computer-writer interaction may clarify CAC-2's questioning routine. Suppose a student is asked to write approximately a page on the topic, "Should students be allowed to choose the subjects they study in school?" As soon as the student has signed on to the computer, the program asks, "Do you have an opinion on this topic?" (If the student answers *no* here, the program encourages the student to consider the questions and formulate an opinion.) When the student answers *yes*, the computer responds, "Okay, let's tell your opinion to the reader," and the student is allowed to enter a sentence. At the termination of the student's sentence, the program halts the text collection and asks, "Would you always believe this no matter what happened?" in an attempt to get the student to clarify and elaborate on the opinion statement. If the student answers *yes*, CAC-2 continues by asking, "Do you have a reason for your opinion?" The program becomes quite stubborn at this point. When the student does not have a reason, CAC-2 indicates that the student must have a reason to continue, and waits for him or her to develop one. Once the student indicates that s/he had a reason, CAC-2 encourages the student to include it. The program continues in this manner, leading the student through questions designed to encourage clarification of the reason, evidence and support for the reason and opinion, inclusion of more reasons, and, finally, a summarization of the paper.

Since there was no way of guaranteeing that the question branching routine would accurately direct the students, the interaction could not remain entirely compulsory. If, for example, the student included both an opinion and a reason in the first sentence, then the question, "Do you have a reason for your opinion?" is inappropriate. In this case or in the case where students simply did not understand the question, they were encouraged to press the "continue" key, and the program would then skip to the next question, assuming that the previous question had been answered favorably. At other times, the student could press the "hold" key, which returned the control of the keyboard to the student and allowed another sentence to be included before the ques-

tion that was on hold would be resubmitted. This option, of course, allowed students to write as many sentences addressing the preceding question as they wished.

In an attempt to investigate the effects that CAC-2 had on students' writing, we tested the program with 36 eighth-grade students. The study involved writing in three conditions, and the results, in terms of the rated quality of the essays, varied depending on the order in which the conditions were presented (Woodruff, Bereiter & Scardamalia, 1981-82). The first condition had students produce a paper-and-pencil baseline essay. On a following day, half the students composed another essay using a version of the CAC-2 program that did not ask any questions, while the other half used CAC-2 as it is described above. On the third day, the students were slotted into the opposite condition so that the study was counterbalanced. Students who produced compositions in the order *baseline, no-questioning* and *questioning* obtained approximately the same ratings under all three conditions. On the other hand, students who produced compositions in the order *baseline, questioning* and *no-questioning* obtained significantly lower ratings in the questioning condition. This finding suggested that the students were overloaded if they had to deal with the novelty of a computer keyboard and a questioning routine at the same time, but that this handicap was largely overcome by a single session of keyboard composing.

Subsequent analysis revealed that the arguments produced in the computer-questioning condition were less well thought out, more one-sided and less mature. A similar decline in performance was observed by Black and Wilkes-Gibbs (1982) with writers who were asked to plan aloud as compared to composing in silence. Apparently any such additional task demand diverts mental capacity from the main task and results in writing that appears simpler or less mature. Thus, while the immediate effects are clearly negative, the long-term effects remain to be tested. Presumably, with practice, the disruptive effects of the novel procedure would disappear; indeed, our results suggest that these effects largely disappeared after just one session. However, educational benefits from the kinds of questioning used in CAC-2 would not be expected until after extended use. For it is not the effects of questioning on the immediate composition that we are concerned with, but rather the change in basic composing strategy that would occur if students became accustomed to considering such questions in the normal course of writing.

Although no long-term testing of CAC-2 has yet been attempted, the response of students to the short-term trial suggests that long-term use is at least feasible. Students themselves rated the compositions they produced with CAC-2 as superior to

those produced under the other conditions. In follow-up interviews students also reported that they preferred CAC-2 to normal writing and that they found the questions made writing easier for them. Finally—and most encouragingly—most of the students reported that they never though about the kinds of questions CAC-2 had led them to consider but that they intended to consider such questions in future writing. We can be sure that it would take more than one trial for the questions to become internalized, but the intention speaks well for the likelihood that the questions would become internalized in the long run.

THE COMPUTER AS COLLABORATOR: EXPLORE

I have described one program that represents a consultative form of interaction and another that represents a more directive form of interaction. Our third program, EXPLORE, represents a collaborative form of interaction by sharing the composing task with the student. This program assumes the responsibility of producing the content and linguistic form of the individual sentences, while the student has the responsibility for the structure and style of the paper. Students are shown a variety of sentences that vary according to their structural function and stylistic form. The student's job is to create a text by evaluating and then selecting or rejecting the presented sentences. EXPLORE has a data base of 308 sentences on the issue of whether or not T.V. is a good influence on young people. Half the sentences (154) are on the pro side of the topic, and the other half are on the con side. Each set of the 154 sentences is further divided into cells representing structural elements of an opinion essay—statement of belief, reasons, elaboration and examples, and conclusion. Each cell contains seven sentences, each of which conforms to a particular style: ordinary (called *OK*), weak, exaggerated, unbelievable, unclear, wordy or jazzy.

Once the student has signed on to EXPLORE, the computer asks whether the student would like to 1) start a new essay, 2) add to an old essay, 3) change the essay, 4) type the essay, or 5) read over the essay. When the student indicates that he or she wants to start a new essay, the computer asks the student to select either 1) a statement of belief, 2) a reason, 3) an elaboration, 4) an example, or 5) a concluding sentence. (The student also indicates whether s/he wants the pro or con side of the argument at this step.) At this point EXPLORE retrieves a sentence from one of the indicated cells and presents it to the student along with a style evaluation list. (See Figure 2 for an example of EXPLORE at this stage.)

Figure 2.
An Example Sentence and List of Style Choices as Presented by EXPLORE

Here is a possible sentence. What do you think of it:

T.V. is not exactly the best influence on most young people.

1. Keep this sentence.
2. It's too weak.
3. It's too exaggerated.
4. It's unbelievable.
5. It's unclear.
6. It's too wordy.
7. It's too jazzy.
8. I want to choose another sentence type.
9. Read over essay.

Type the number ?

If the student indicates that s/he wants to keep the presented sentence, EXPLORE adds the sentence to the paper and branches back to the structural element selection routine. On the other hand, if the student rejects the sentence as being too much of one style or another, the computer selects an alternative sentence for presentation. This selection is dependent upon the student's evaluation of the previous sentence. For example, if the student evaluated the previous sentence as too unbelievable, and indeed the sentence had been authored to be of that style, then EXPLORE presents the *OK* or ordinary version. If the student evaluates the previous sentence incorrectly, then the next presented sentence is selected to contrast with the last evaluation. For instance, the student may call a sentence too weak, when in fact it was authored to be the *OK* version, so the computer would select the exaggerated version for presentation. The student continues in the above manner, selecting and rejecting sentences until she or he feels the essay is complete.

EXPLORE's contrasting attribute was also included in its editing options under the "translate" function. EXPLORE's editor allows the student to 1) change the order of the selected sentences, 2) delete sentences, 3) read the sentences with reference numbers, 4) translate a stored essay into a requested style, and 5) restore a stored essay to its original version (i.e., return the essay to the state it was in when the student entered the editor). The translate function is quite unique in that it allows the student to take a previously produced essay and translate it into any one of the seven different styles. This option allows the student to see, with ease, the rhetorical effects of style while the structure and content of the paper are held constant. (See Figure 3 for two examples of this function.)

EXPLORE is designed to ease the mental burden of composing. Since the computer is producing the content and linguistic form of the text, students are able to produce more essays per class than they would be able to under normal conditions. With this facility students are allowed to concentrate on aspects of the composing task that are seldom made conscious. Furthermore, students are encouraged to experiment with novel forms of structure and to evaluate the relative effects of style. And it is through such efforts that we expect the users to construct some of the higher, more sophisticated, composing strategies.

Figure 3.

Three Sample EXPLORE Texts Showing the Original and Translated Jazzy and Wordy Styles of the Original

ORIGINAL

1. T.V. is a bad influence on young people.
2. Too much violence is shown on T.V.
3. The overemphasis on death and bloodshed has made violence an almost acceptable way of life.
4. Murder trials have recently taken place where the defendant pleaded T.V. addiction.
5. For these reasons, it is clear that T.V. is a poor influence on young people.

JAZZY

1. T.V. is the real pits.
2. Bang! Pow! Thud! Aargh! That's what T.V. is all about.
3. People don't freak out about crime anymore, because they see so much of it on T.V.
4. If you see somebody getting mugged on the street, you stay cool and say, "So what else is new?"
5. So, now you can see that T.V. is the real pits, and you'd better believe it.

WORDY

1. Watching T.V. is definitely a bad influence on the young people of our modern world of today.
2. Television is a bad influence on our people because it shows far too much violence that is not healthy for young people to watch.
3. Because of people watching T.V. so much, they get used to what they see on T.V. and think it is normal, which is violence.
4. For instance, if you see a lot of robbery and shooting and kidnapping, etc., on T.V., then you start to think it is normal

to have robbery and shooting and kidnapping, etc., in your own city where you live.

5. If you consider these facts and points, I am sure you will conclude that, as I said at the beginning, T.V. is a bad influence on the young people today.

Exploratory tests of the program on grade 12 students found some evidence suggesting that students will experiment with the style of the sentences and the structure of the text. This data, collected over eight successive classes, indicated that the distribution of sentence styles tended, in the EXPLORE essays, to be less weak, unclear and exaggerated, but more wordy than students would normally write (Bereiter, Scardamalia & Woodruff, 1982). This experimentation suggested that students were exercising skills they seldom have the time to concentrate on. One of the findings, as evidenced by the students' post-experimental impressions of the program, was that students increased their awareness of text structure. The interview data, collected from one of the classes involved in the study, indicated that over 90% of the students felt their experience with EXPLORE taught them something about the structure of a paragraph. A clear example of what most students felt they learned is extracted from one student's text and presented below:

> What I did learn, in fact, from the program was how to visualize an essay as a set of individual points that need to be correctly arranged in a sensible order, like constructing a building. The bricks of a building need to be placed properly in order to form a solid foundation, this is also true for an essay.

As a collaborative program, EXPLORE was a success. It did not try to move students through great leaps and bounds, but it did give them the opportunity to practice some of the skills that would normally have been difficult to exercise. The degree of EXPLORE's success is dependent upon the classroom teacher. The program is most successful when it is combined with challenging tasks. One such task is to give the students the goal of trying to make their papers sound as if they had been authored by someone else such as an expert, someone genuinely concerned, someone overly emotional about the subject, or a pompous windbag. Another task is to convince different types of audiences. Alternatively, the teacher may wish to see the students manipulate the structure of the texts by giving them the goal of making one point, but making it clear, or of trying to balance arguments on both sides of an issue. The number of interesting tasks is limited only by the imagination of the teacher, and the more interesting the tasks, the more successful EXPLORE is likely to be.

CONCLUSION

In discussing EXPLORE, CAC-1 and CAC-2, I have presented three different modes of interacting as built into three different programs. The least intrusive program, CAC-1, simply acts as consultant. While this program has the advantage of helping to reduce the writer's mental burden, it has the disadvantage of being incorporated readily into a "What next?" composing strategy. CAC-2, on the other hand, can override the "What next?" strategy, but it appears to do so, at least during initial use, by dividing subjects' attention among more task features than they can handle as they compose. Finally, our last program, EXPLORE, demonstrates one way of dividing up the composing process so that the student and computer can work collaboratively. This form of interaction appears to be best suited to situations where dividing the task affords the opportunity to develop mature strategies. EXPLORE concentrates only on selected aspects of the composing process, but it allows greater forms and amounts of exercise than can be experienced under normal conditions.

It seems clear that different forms of interaction influence different aspects of the composing process. In our current work we are experimenting with a program that interacts with the student during planning rather than during writing a compostion; yet another program we foresee is one that will provide procedural support to students as they evaluate, diagnose and revise a previously written composition. As I noted at the beginning, the possible roles one may design for a computer are limited by the computer's capabilities and by our applicable knowledge of the composing process. But both of these areas are advancing rapidly. As computers become more sophisticated at interpreting language input and as we become more sophisticated in our understanding of what goes on in the minds of students when they write, we can expect to see subsequent generations of interactive programs that help student writers in increasingly educative ways.

REFERENCES

Bereiter, C., & Scardamalia, M. From conversation to composition: The role of instruction in a developmental process. In R. Glaser (Ed.), *Advances in Instructional Psychology* (Vol. 2). Hillsdale, NJ: Lawrence Erlbaum Associates, 1982.

Bereiter, C., Scardamalia, M., & Woodruff, E. *Functional and stylistic choices in computer-assisted instruction.* Paper presented at the annual meeting of the American Educational Research Association, New York, March 1982.

Black, J.B., & Wilkes-Gibbs, D. *Distortions of the writing process by verbal reports of mental processes.* Paper presented at the annual meeting of the American Educational Research Association, New York, March 1982.

Woodruff, E., Bereiter, C., & Scardamalia, M. On the road to computer-assisted compositions. *Journal of Educational Technology Systems,* 1981-82, *10*(2), 133-148.

PROGRAM INFORMATION

Explore

Explore is now a part of the courseware used on a Honeywell system at the Ontario Institute for Studies in Education. For more information write to Dr. William Olivier, Ontario Institute for Studies in Education, 252 Bloor St. West, Toronto, Ontario M5S 1B6.

WRITER'S HELPER: A System Approach to Computer-Assisted Writing

by
William Wresch
University of Wisconsin Center-Marinette

I have been intrigued for a number of years with the idea of using computers to help with prewriting. Obviously I am not alone. Hugh Burns has been working for years on his heuristic questions, Helen Schwartz of Oakland University has developed computer-assisted prewriting programs for literary analysis, while Dawn and Ray Rodrigues have developed a whole series of prewriting programs at New Mexico State. I suspect there are many more good, if less publicized, prewriting programs around.

The direction I have chosen to take with my efforts has been to combine prewriting programs with writing and revising programs. This is not the approach I began with. Two years ago I designed a program called *Essay Writer.* It was a program which asked students a series of questions about a subject and then created a five paragraph essay for the student by manipulating the student's answers. (For a more complete description see *The Computing Teacher,* March, 1983.) What was produced was a passable short theme, but one which needed a fair amount of revising. And there was the problem. Without a connecting word processor, the essay had to be totally retyped before it could be modified by one of the commercial word processors available. Students weren't too anxious to do that. I wasn't surprised.

I decided it was time to try to create a program which would help with prewriting, supply a word processor for writing, and

while I was at it, why not a few programs to do an initial analysis of students' work. The result is a program I call *Writer's Helper.* It is actually a dozen or so programs linked together on a single disk.

By far the greatest number of programs deal with prewriting. Here my approach was to listen to the advice of Ray and Dawn Rodrigues who suggest multiple prewriting activities so students have a choice, and to follow my own experience which told me that students have different levels of need, with some students unsure about even where to begin, while others have a sense of direction but need help along the way.

There are three programs in *Writer's Helper* to help students find an initial subject. One, *Brainstorms,* is a computerized freewriting activity which has a timing feature that prints an "X" if a student doesn't type anything for over two seconds (the length of time can be set for varying groups). So if students do what so many do and halt for a time, the program slowly prints a series of X's to get them started again. The program also keeps track of how much a student actually writes during the time set, and advises the student about how well his or her volume of output matches the intent of the activity.

Another program in this series uses list generation to help students find a subject. The third program asks a series of questions to help the writer find a subject. Most are along the lines of "Whom do you most admire?" and "What makes you proudest of your school?"

The next group of three programs is designed for people who have a general idea for a subject but haven't looked at it closely enough yet to be sure what to say about it. I spent a great deal of time on these three programs because this is the level where most of my students have their trouble. They either can't go beyond an opening statement, or they can't find an opening statement which isn't unbearably trite (e.g., "Cross country skiing is lots of fun.").

The first program in this group, *Crazy Contrasts,* asks students to type in a short subject, and then generates a series of contrasts. For instance, if a student had typed in "cross country skiing" for a subject, the program might begin as follows: "Type in three ways cross country skiing is like day-old meat loaf," or "Type in four ways cross country skiing is similar to Richard Nixon." The contrasts are meant to be humorous, but they can make the student think about a subject in new ways. I for one can think of several ways essays on cross country skiing are like day-old meat loaf.

The second program asks students to identify a subject and then list twenty related facts or ideas. Having done this students

are shown an example of categorization, asked to categorize their own list into three to five groups, and then the program automatically produces a tree graph of the student's subject, categories, and list items. The purpose, of course, is to help students define an idea by seeing its subheadings.

The last program at this level contains more questions designed to help the student see his or her idea in isolation, as part of a process, and as part of a network—in short, my poor effort at imitating Becker, Pike and Young. As students progress through the questions they are told about the particle, wave and field approach and asked to identify for themselves which is most appropriate for their subject.

I established my next level of programs for those who have progressed through the first two levels and now have a clear subject and an approach to that subject, but aren't sure how to organize their ideas into an essay. This level is essentially a reworking of my old *Essay Writer* program changed to fit its now more restricted use, and to respond to criticisms that *Essay Writer* ignored audience.

The program begins with a brief description of its intent, and then asks the user to type in a subject. If you will allow me, I'll type in the subject "Computer-assisted writing" and follow the program from there. My responses will be in bold face type.

What form your essay takes will depend upon your purpose. Several such purposes are listed below.

1) Describe the appearance of computer-assisted writing.
2) Compare computer-assisted writing to another subject.
3) Show how computer-assisted writing has changed over time.
4) Describe the job or function of computer-assisted writing.
5) Show what caused computer-assisted writing to be the way computer-assisted writing is now.
6) Describe the importance of computer-assisted writing.

Type in the number of the purpose which is best for your essay and press RETURN.

3

So you are going to organize an essay around the changing of computer-assisted writing.

This can be done many ways. Let's start by thinking about your audience. Type in a word or phrase naming your most likely audience. Remember to press the RETURN key when you are done.

Audience: **English teachers**

Good. Now let's think a little about what kinds of opinions English teachers already have about computer-assisted writing.

If you asked English teachers about computer-assisted writing,

which of these statements do you think they would most likely agree with?

1) Computer-assisted writing is very much changed.
2) Computer-assisted writing is not very changed.
3) Computer-assisted writing is changed in some surprising ways.

Type in the number of the statement you think you audience would most agree with and press RETURN.
1
What do YOU think? Do you agree with your audience? Type in a 'yes' or a 'no' and press RETURN.
no

Good. You can explain to your audience why they are wrong. You may have to supply some convincing proof, but this should be an interesting essay for them to read.

Start by listing 3 ways in which computer-assisted writing is not very changed. Remember to press RETURN after each one.

1) **We still use the questioners of Ellen Nold.**
2) **We still analyze features of style much as Ellis Page did.**
3) **Our word processors have only a few features different from those of 10 years ago.**

Now let's describe those ways in which computer-assisted writing is not very changed.

The ways you listed were:

1) We still use the questioners of Ellen Nold.
2) We still analyze features of style much as Ellis Page did.
3) Our word procesors have only a few features different from those of 10 years ago.

To begin with, type in 4 complete sentences which help explain what you meant when you said We still use the questioners of Ellen Nold. Press RETURN after each one.

1) **Ellen Nold wanted to use the computer in prewriting by having it ask a series of questions.**
2) **Hugh Burns was one of the first to follow her lead.**
3) **Helen Schwartz uses her approach in analyzing literature.**
4) **I constantly find myself asking the same questions she wrote about in the mid 1970s.**

Good.

Let's try the next paragraph. In this paragraph type in 4 complete sentences which explain what you meant when you said We still analyze features of style much as Ellis Page did. Press RETURN after each one.

1) **Ellis Page looked at word length and sentence length and word type back in the mid 1960s.**

2) **The HOMER project at UCLA looks at many of the same things.**
3) **So does Writer's Workbench.**
4) **We are just doing what Page started nearly twenty years ago.**
4) ?We are just doing what Page started nearly twenty years ago.

Good.

Let's try the next paragraph. In this paragraph type in 4 complete sentences which help explain what you meant when you said Our word processors have only a few features different from those of 10 years ago. Press RETURN after each one.

1) **The standard of a professional word processor is to insert and delete words easily and do automatic searching and replacing of words.**
2) **Ruth Von Blum of UCLA is designing a word processor which will move from sentence to sentence and paragraph to paragraph on the screen but she is the only one looking at such features.**
3) **Most of us are comfortble using the same features but adding in 'MENUS' and such to make them easier for children to use.**
4) **We haven't really added new features that use what we as English teachers understand about the process of writing.**
4) ?We haven't really added new features that use what we as English teachers understand about the process of writing.

Good work. You now have identified three ways in which computer-assisted writing is not very changed and described each of them in detail.

Now let me show you how all of what you told me could be put together to form an essay.

Press the space bar and I will start.

Most people think that computer-assisted writing is very changed. But that really isn't true. Here are three ways in which computer-assisted writing is not very changed.

One way in which computer-assisted writing is not very changed is We still use the questioners of Ellen Nold. Ellen Nold wanted to use the computer in prewriting by having it ask a series of questions. Hugh Burns was one of the first to follow her lead. Helen Schwartz uses her approach in analyzing literature. I constantly find myself asking the same questions she wrote about in the mid 1970s.

Another way in which computer-assisted writing is not very changed is We still analyze features of style much as Ellis Page did. Ellis Page looked at word length and sentence length and word type back in the mid 1960 s. The HOMER project at UCLA looks at many of the same things. So does Writer's Workbench. We are just doing what Page started nearly twenty years ago.

The last way in which computer-assisted writing is not very changed is Our word processors have only a few features different from those of 10 years ago. The standard of professional word processor is to insert and delete words easily and do automatic

searching and replacing of words. Ruth Von Blum of UCLA is designing a word processor which will move from sentence to sentence and paragraph to paragraph on the screen but she is the only one looking at such features. Most of us are comfortable using the same features but adding in 'menus' and such to make them easier for children to use. We have not really added new features that use what we as English teachers understand about the process of writing.

So there shouldn't be much doubt. Because We still use the questioners of Ellen Nold and we still analyze features of style much as Ellis Page did, and our word processors have only a few features different from those of 10 years ago, it is clear that computer-assisted writing really is not very changed.

How's that for an essay? Pretty rough, isn't it? But it does show you one way your ideas about computer-assisted writing could be organized. Press the space bar to continue.

If you'd like, I can take this rough essay and save it onto a disk so it can be transferred to the word processor to be fixed up or printed out on paper.

Once saved to disk, the *Writer's Helper* word processor can be used to add or delete any portions of the essay that need to be changed, and of course that is where all typographical errors could be corrected.

Once the essay has been saved, the program goes on to explain the process it used in developing the essay, and then urges the student to rewrite the rough draft the program has produced. At this point students should have seen one way in which their ideas can be organized, and should be ready to try a more creative approach.

The last of the pre-writing programs works with individual paragraphs and is really more intended as a tutorial for students who are having a hard time with a single paragraph in an essay. It describes both descriptive and expository paragraphs and demonstrates how they can be created from details, examples and statistics.

The next large program in the *Writer's Helper* group is the word processor. I felt I wanted a word processor to make *Writer's Helper* a complete system, and there were several features I wanted but couldn't find on commercial word processors, but let me say in advance to others who are considering writing a word processor, it is five times more difficult than you may imagine. I hired one of the best computer science students on campus, David Kraus, and did much of the initial writing and designing myself, and even with the two of us working on it, the job took all summer.

Ultimately what I got was a processor that worked well with all the other programs, had complete upper and lower case so students didn't have to go through some of the inconveniences of *Apple Writer*, maintained a menu of alternative actions across the top of the screen so no one had to memorize which combination of keys caused a deletion or allowed block moves, and that was it. I had originally toyed with all kinds of wild schemes —keys that printed words out in phrases, cursor jumps from noun to noun, and automatic inverse printing of passive verbs. Oh I was full of ideas, but as August came to a rapid close we decided to go with what we had.

The last set of programs I wrote were intended to help a student re-evaluate an essay or report. I did this work last and have to admit it was the part that interested me the least. I know there are many English teachers who use analysis programs such as HOMER and *Writer's Workbench* and have great success with them. I also know there are many who are scared to death that setting automatic standards for style will stifle creativity. We can all guess what *Writer's Workbench* would say about "Hamlet" or the Declaration of Independence. I finally decided to go with the first group, not because I want computers to do my essay grading for me, but because I saw that the *Workbench* and HOMER people were using those programs to call students' attention to style. For them such programs are a way of getting students to think about writing. Students who use HOMER may or may not take the computer's opinion of their passivity seriously, but at least they know what the passive voice is and where it occurs in their own writing.

So I put together some analysis programs. Four are intended to help students examine their own style, while two are there to save me work. The best example of the latter pair is a program to check on homonym errors. If I had a dime for every "to, two and too" confusion error I've seen I could finally get new tires for my car. So I designed this program to search through any essay a student has written in the *Writer's Helper* word procesor and find all the homonym problems present. In the case of "could of" the program points out the error. In the case of "there, they're and their" the program defines the correct use of each and asks the student if the use underlined by the program is correct.

The usage checker program operates on the same basis. It finds potential trouble words in a student essay, defines correct usage, and leaves it up to the student to determine whether an error exists.

The four style programs start with one program which points out the first sentence of each paragraph of a student's essay and then describes what cohesion should exist to demonstrate clear organization.

Another program graphs out the length of each sentence of a student's essay and describes how the graph can be interpreted.

A similar program graphs out paragraph lengths and calls attention to shorter paragraphs, suggesting causes for possible underdevelopment.

The last program in this group is a readability analysis. It asks students to estimate the grade level of their audience and then matches that projected grade level with the calculated grade level of the student's essay. Should the student's essay be much above or below the projected level of the audience, the program describes some of the causes for high or low readability.

As in other analysis programs, these programs are intended only to start students thinking about their writing. I don't think any of us are prepared to let a machine set some arbitrary standard for us.

And that's it—slightly more than a dozen programs which are intended to help students during most of the writing process. All of them could be improved, and we could all think of others to add, but these are at least a start. If they have any overriding virtue it is that they are programs designed to be an active part of the writing process, and that they function well as a whole, with students able to move their essays from one program to another as they wish. At least until next summer vacation, that is where *Writer's Helper* is going to have to remain.

PROGRAM INFORMATION

System Requirements:

Writer's Helper uses an Apple IIe with 80 column card and one disk drive (what Apple calls its "start-up system"). A printer is recommended, but need not be attached to every computer using *Writer's Helper.*

Program Availability: *Writer's Helper* is currently being evaluated by several high schools and colleges. It should be available commercially in the spring of 1984.

Courseware Design

Design Considerations

by
Alfred Bork
University of California at Irvine

One of the problems in education is that we are much too concerned about teaching and not enough concerned about learning. So I'll concentrate on learning.

BAD PRACTICES

I want to start with a brief list of bad practices in computer-based learning. As developers, we must be aware of such practices so that we can avoid them in our materials. If you look at materials generally, you will find that these bad features are very common.

Screen Design

The first set of issues in my compendium of bad practices deals with screen design—particularly hindrances to readability on the screen. We have to remember that blank space is free on the computer, unlike print media. Many designers come into this field with a book mentality, and they tend to show things the way they would look in books. For example, long lines are something we don't need in computer-based materials. Readability research gen-

erally shows that short lines aid readability. We sometimes see words being split arbitrarily at the end of the screen. That should never happen in a student-usable program. A similar problem arises when users input text. Generally, the input does not use "word wrap" techniques. Developers of word processing systems learned long ago that this is a poor practice. It's easy to build software that automatically moves complete words to the next line on the screen.

We also frequently see computer dialogues putting almost all text at the left margin—another print technique. Many computer languages assume that you will always want to put text at the left margin. But when you start using more blank space, you will want to balance material on the screen, and you will seldom put anything at the left margin. More generally, there isn't much conscious placement of text in many materials that exist today. Most people are unfortunately just letting the system put text wherever it happens to put it.

Another thing we see frequently is splitting natural phrases across a line. Studies again show that this is a deterrent to readability. Given that blank space is free on the computer, there is no reason that we can't display text logically, keeping natural phrases together on a line.

In addition, much of the text we see is displayed too fast. Our studies have shown that novice users don't like this rapid text display. Given a choice of where to set the speed of text, such users set it at a much slower rate. Good programs will allow users to set the rate of the text display. This is not difficult to build into the software.

Timing

Timing is another important issue in the design of computer-based materials. In most environments we cannot assume that people are going to be there forever. If a user doesn't know that s/he needs to press RETURN (or some other key) to continue, the machine should tell the user to press RETURN. In addition, there may be instances when you need to time the student's response. But this has to be done in a sensitive fashion. That is, if a student waits nine seconds and then starts to type, we don't want to cut him or her off just because we were waiting ten seconds.

Content

Next, let me look at a few content issues that are also in my compendium of bad practices. First, it seems fair to say that the vast majority of the programs currently available are trivial programs. That is, they are not really serious curriculum efforts.

This abundance of trivial programs has probably occurred because we are at a very early stage of development. As we gain more experience in designing computer-based materials, we should start moving away from trivial applications.

Another issue related to content concerns "help" sequences. For the most part these sequences are not very helpful. They are usually completely noninteractive. Often a help sequenceis simply a full screen of verbal material to read. Help sequences can be just as interactive as the rest of the material.

Media Issues

The fourth category in my compendium of bad practices includes media issues. Most computer materials we see today look like a book. Part of this comes from the fact that there is not enough visual material. Instructors are extremely verbally oriented, and they often do not understand the visual needs of students. As many of you have undoubtedly discovered, not all students are verbally oriented.

Another important media issue is solving design problems with other media. I get very worried when I am making a film and somebody brings up a problem and a designer says, "Oh, we'll put it in the student handbook." I get very worried when someone designing a computer program is asked a question about it and says, "Oh, that's covered in the teacher's guide." I get suspicious when designers have to solve problems with other kinds of media.

Another major media issue is the amount of interaction. The real advantage of the computer is that it is an interactive medium. However, most teachers tend to come from successful environments of lecturing, where they don't think and work interactively. Consequently, computer-based materials tend to look like books. Much more interaction is needed.

General Issues

In any interactive learning material, we ought to use English as the medium of communication. This business of using *Y* and *N* and special letters as responses seems to be a questionable strategy. The English language is a perfectly reasonable tool to use, and I don't think we need "computerese." If we don't use good form in the material, how can we expect students to use good form?

Another issue is the use of multiple-choice formats. Multiple choice came about as a tactic of desperation for dealing with large numbers of students. But we don't need multiple choice on the computer. This isn't to say that we might not occasionally want to use it just for variation, but as a general rule, multiple choice is a terrible strategy.

DEVELOPING MATERIALS

I would like to turn now to a discussion of the principles that should guide our development efforts. It makes sense to begin by examining the strengths of the computer as an instructional medium.

First, the computer is an interactive medium—and interaction is not just a word. In fact, I believe that we can measure interaction. We can look at computer material and talk about the *quality* of interaction. Interaction doesn't just mean that the user gets to say *yes* or *no* every few minutes while reading. We can look at how frequently interaction occurs, we can look at the types of interaction, and we can begin to arrive at measures of interaction.

The second major strength is the computer's capability for individualization. If computer material is good, it can really begin to do what we've always talked about in education—individualize the learning experience. We can make the learning experience different for different people.

Another strength of the computer is that it can present us with new ways to organize learning. This is probably the hardest thing for the conventional teacher to understand. Often the computer allows entirely different strategies for organizing our material.

The Development Process

Development is a serious process—not something that can be done casually. If one is going to develop learning material of any kind, one doesn't do it by finding a teacher and letting that teacher do it in his or her spare time. The notion that we can easily produce good learning material is an illusion.

We need groups in serious development efforts. We can't assume that a good teacher is going to be a good programmer, just as we can't assume that a good teacher knows how to run linotype machines and printing presses if we are developing textbook material. We need to look at the places where serious curriculum development takes place. My favorite example is the Open University in England. The Open University typically spends a million dollars on developing a course these days. The University runs the course for seven years and then scraps it and spends another million dollars. We are not going to get good computer-based learning material until we also begin to develop at that level.

Instructional designers ought to stop thinking about programming. For the most part, the people who write textbooks don't learn how to run printing presses. And the people who are going to design good computer-based learning material are not going to be computer experts. Programming is a technical area,

and as we get into commercial production, where issues of transportability to new machines are extremely important, we are going to have to separate instructional design from technical production, much as we do now with books, films, and almost every other kind of learning medium.

Research Needs

Perhaps the most important issue in developing computer-based materials is establishing a research basis. For example, this has been a conference on writing. What do we know about writing? What is the experimental evidence? What research do we need to conduct in order to design successful materials in this area? Who are the experts in teaching writing, and who are the experts in research associated with writing?

Bob Shostak has already noted the problems in teaching writing. This seems to be a very solid place to begin our development efforts. We shouldn't start by saying, "What can I do with computers?" Rather, we should start by asking, "What are my pedagogical difficulties? What are the problems associated with teaching writing?" It may turn out that computers are going to be extremely useful (as I believe they will be) in dealing with these problems.

The Future

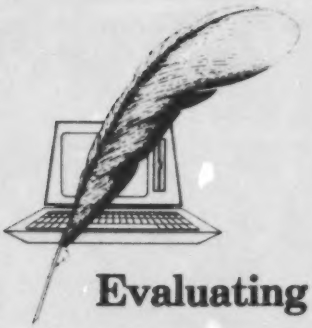
As developers, we need to have a very strong orientation toward the future. The problem is not what we can get into classrooms tomorrow. The really interesting things are future-oriented. We are at the threshold of a real revolution in the way people learn—a revolution that is going to influence our entire educational system. We ought to look very much into the future, and there are several implications to this orientation. For example, if you are a developer, you don't want to develop materials for the kind of hardware that is currently available. Such machines are likely to be obsolete within a few years.

We also need to think more about non-school environments. Too much of our thinking about computers concentrates on schools, and we often hear people say, "Will teachers accept this?" In fact, teacher acceptance may not be very critical in this area. As computers become more common in homes, learning material will become an important component in selling computers. Consequently, commercial interest in learning will be great, and our whole attitude toward education will change.

I would like to raise an additional issue for you to think about. I have already pressed the notion that the computer can really lead to new ways of organizing instruction, new ways of doing things. Now I raise the question of whether reading and writing

really are separate activities. Or are they really things that can be done together on the computer, rather than being done separately? Interestingly, the most extensive early work involving the computer and typewriters, O. K. Moore's "talking typewriter" study, was really a reading project rather than a writing project. This early work showed that you can teach very young children to type, and that you can use typing as a vehicle for teaching reading. If we begin to think about the computer as a "combining" device, then perhaps we can approach reading and writing as fundamentally the same activity.

Finally, I always like to remind people that it is not clear that the computer is going to improve education. The computer, like any new technology, has the potential for improving education or weakening education. Everything in human technology has this dual potential. When humans first learned to use fire, they found that fire could be used for good or for bad. The technology itself doesn't determine whether it is going to be good or bad. People do. And we have to remember that the computer is a gift of fire.



Evaluating Textual Responses

by
Joseph Lawlor
SWRL Educational Research and Development

Under a contract with the National Institute of Education, my colleagues and I here at SWRL have been investigating the use of microcomputers to teach writing. The purpose of our investigation has been to explore ways in which the computer might be useful for teaching the high-level skills that are associated with composing.* Although much of our work has been exploratory, our experiences in designing interactive writing materials may be of interest to developers—or potential developers—of computer-based writing instruction.

When we began our project, one of our first tasks was to find out what kind of instructional programs were currently available for teaching writing. Not surprisingly, we found the field to be very limited. Most of the available courseware was of the drill-and-practice variety, dealing primarily with the form skills of writing such as spelling, usage and capitalization. Although such programs can be useful, very few of them require the student to write anything longer than a single word. In fact, since many of the programs employ multiple-choice formats, the student commonly responds by simply typing a single letter or number.

However, when students write, they are expected to produce sentences and paragraphs, not single words or letters. Consequently, we decided to focus our efforts on designing computer-based materials that would require the actual production of text.

---

*Additional information on this project can be found in "Using Microcomputers for Composition Instruction," by Bruce Cronnell and Ann Humes (ERIC Document Reproduction Services No. ED 203 872).

And we also felt that our programs should be able to evaluate that text.

After exploring a number of ideas for instruction, we decided that our initial effort would be to design a program for teaching sentence combining. There were several practical reasons for this decision. First, sentence combining has been shown to be an effective technique for improving the sentence structure of students' writing. Second, after reviewing the research on sentence combining, we had a pretty good idea of what the scope and sequence for such a program should be. And finally, we felt that while sentence-combining instruction would allow us to work with textual responses, it would also allow us to limit the range of those responses, making the evaluation task more manageable.

EVALUATING STUDENT RESPONSES

As the framework for our sentence-combining program began to take shape, we soon discovered that evaluating students' responses was not going to be an easy task. The computer—for all its remarkable electronic sophistication—hasn't the faintest idea of what a sentence (or even a word) is. When a computer stores text in its memory, it stores the text simply as a "string" of characters, that is, a particular sequence of letters, spaces and symbols. Although the machine does not "know" what the string means, it can compare one string to another and determine whether or not the two are equal (i.e., whether or not the strings contain exactly the same characters in exactly the same sequence). Consequently, the easiest way to evaluate the student's response would have been to compare the student's response to the correct answer. If the two strings were not equal, the machine could then tell the student that his or her answer was wrong and branch to some type of remedial instruction.

However, we felt that there were degrees of "rightness" and "wrongness" that had to be considered in evaluating the student's response. For example, in the following item a student who had correctly combined sentences (1) and (2) would produce sentence (3):

(1) Tom fell to the ground.

(2) Tom ripped his uniform. (and)

(3) Tom fell to the ground and ripped his uniform.*

---

*This item format was originally developed by Frank O'Hare (*Sentence Combining: Improving Student Writing without Formal Grammar Instruction.* Urbana, IL: National Council of Teachers of English, 1973). The parenthesized word is a "signal" that tells the student how to combine the sentences In this case, the student is to conjoin the two predicate phrases with *and.*

But consider the following hypothetical responses:

(4) Tom fell to his uniform.

(5) tom fell to the grond and ripped his unifrm

To the computer, both of these responses would look equally wrong; neither of them would match the correct answer. Yet there are important differences between the two responses. A student who enters the first sentence evidences little or no understanding of the sentence-combining instruction. But a student who enters the second sentence seems to comprehend the material; errors in the second sentence reflect spelling, capitalization or typing problems—not problems with sentence combining.

Clearly, a simplistic evaluation of the student's response was not appropriate for our sentence-combining program. What we needed was a system for "parsing" the student's sentence—a system that would examine the sentence and distinguish among the various types of errors a student might make.*

DESIGNING THE PARSER

We began specifying the functions of our parsing system by classifying the errors that students might make into two broad categories. The first we called "syntax" errors—errors that reflected a misunderstanding of the sentence-combining task, as exemplified by response (4) above. The second category included errors of form (spelling, typing, capitalization, punctuation), as illustrated by response (5). A student whose response contained a syntax error was to be routed back through remedial instruction. However, a student whose sentence contained only errors of form was to be allowed to fix those errors until the sentence matched the desired response.

Several of the parsing routines were relatively easy to specify. As it is now written, the program first checks for a capital letter at the beginning of the student's sentence. If the first letter is not a capital, the program asks the student to make the necessary correction. If the student does not change the letter at this point, the program will automatically make the correction. A similar routine checks for the presence of a period at the end of the sentence.

---

*The term *parsing* as used here does not refer to formal grammatical analysis. Although computer programs have been developed that can perform such analyses, these programs normally require extensive amounts of computer memory—more memory than is available in the microcomputer with which we were working. For a review of recent developments in the field of computer-based grammatical analysis, see Hendrix, G., and Sacerdoti, E., "Natural Language Processing: The Field in Perspective," *Byte*, September, 1981.

The next parsing routine checks for the presence of the sentence-combining "signal" in the student's sentence. If the signal is not present, the program asks the student to add it. At this point, if the student does not add the signal, the program detects a syntax error and branches to the remedial instruction.

Next, the program counts the number of words in the student's sentence. (A word is defined as a series of letters followed by a space, or, in the case of the last word in the sentence, followed by a period.) If the number of words in the student's sentence does not correspond to the number of words in the correct response, the student is informed of the discrepancy and asked to revise his or her response. Failure to revise the sentence correctly results in a syntax error and a branch to remedial instruction.

At this point in the evaluation—if the student has not been routed back through the remedial instruction because of a syntax error—the student's sentence now contains an initial capital letter, a terminal period, the sentence-combining signal, and the correct number of words. All that remains now is to check for spelling errors. Although checking for such errors would seem to be a relatively straightforward task, we found that this was actually one of the most complicated routines in the whole program. We wanted the spelling routine to examine each word in the student's sentence, compare it to the corresponding word in the correct answer, point out any word that was misspelled, and ask the student to correct the misspelling. We also wanted the routine to include a spelling corrector—that is, if the student did not correct the word on the first attempt, we wanted the program to correct the word automatically, so that students would not get bogged down in endless editing tasks, which would contribute little to the primary purpose of the instruction.

The major problem we encountered in designing this routine was defining a spelling error. Consider, for example, the following response to the item listed above. Both of these responses would have passed through the previous parsing routines:

(6) Tom fell to the ground and he fell down.

(7) Tom fell to the grond and ripped his unifrm.

Obviously, the errors in sentence (6) are not spelling errors. But to the computer they might appear to be. For example, when the machine compares the seventh word in the student's sentence *(he)* to the seventh word in the correct answer *(ripped)*, the computer will detect only that the two words are not identical. Without additional programming the computer cannot tell whether the student's word is misspelled or whether it is simply the wrong word. Consequently, we had to develop an additional parsing routine to

examine each word in the student's sentence, letter by letter. We then defined "spelling" errors as discrepancies that resulted from one of the following conditions:

- The word contains one letter that is different from the corresponding letter in the correct word (e.g., *grount* for *ground*).
- The word contains one letter too many (e.g., *geround*) or one letter two few (e.g., *groun*).
- The word contains one pair of transposed letters (e.g., *gruond*).

All other discrepancies in the student's text are treated as syntax errors. Thus, when the computer encounters the word *he* in sentence (6), the student is sent to the remediation routine. However, the discrepancies in sentence (7) are treated as spelling errors: The student is asked to correct the errors, and the program then proceeds to the next activity in sequence.

CONCLUSIONS

Our experience in designing the sentence-combining program has led us to form several conclusions about computer-based composition instruction—conclusions that are likely to have important implications for subsequent development efforts in this field. First, designing interactive programs that evaluate textual responses is not a simple task. Programs that require only minimal response from students—entering a single word, a letter or a number—can evaluate those responses quickly and easily. However, extended textual responses demand more complex evaluation. Even in our sentence-combining program, where student input is reasonably constrained, evaluation must distinguish among many different types of errors, responding to each type in a different way. Imagine the complexity that would be involved in evaluating less constrained responses, for example, a narrative paragraph that must be checked for appropriate transitional expressions.

Another conclusion we have drawn from our experience is that no matter how sophisticated our evaluation techniques become, they will still not account for *all* the responses that students might possibly make. The best we can hope for is a reasonable compromise. An evaluation routine has to be flexible enough to handle a variety of error types, but it must not be so flexible that it allows faulty input, such as typing *grated* for *ground*. In evaluating textual response, we have to remember that there are definite limits to what we can do.

Finally—and perhaps most importantly—we have found that the extra effort required to design sensitive evaluation routines pays off in the long run. Students who participated in pilot studies of our sentence-combining program seemed to appreciate the way their errors were handled. Most of the students were very willing to revise their sentences once the computer provided the appropriate prompt. Students tended to view the computer as a partner rather than as an adversary, and, after all, that is precisely what a good composition teacher should be.

A paper describing the developmental work of the sentence-combining program is available by writing Joseph Lawlor, SWRL Educational Research and Development, 4665 Lampson Ave., Los Alamitos, CA 90720. The program was written for an Apple II or Apple II+ and requires two disk drives and an Apple clock.

Developing Interactive Language Experiences for the Computer

by
Gloria Kuchinskas
Boca Raton, FL 33432

The metaphor of the computer as a pencil is one Seymour Papert uses to describe his view of an electronic educational future (Papert, 1981). The pencil is universal; it never dictates what we do with it; it enhances our ability to do anything we want to do with it. When students are allowed to treat the computer as if it were a pencil, then the computer and the student will react with each other. For microcomputers to become a real tool in the language arts classroom, teachers must accept this metaphor.

In discussing activities in artificial intelligence—the attempt to make computers smarter—Krutch (1981) uses the term "natural language processing" to describe the same approach to developing interactive language activities for the computer; several authors are also emphasizing the use of the computer as a language motivator or generator.

Four basic educational principles should be kept in mind when we seek to develop interactive language programs for the computer. One is that in talking about language we are talking about production—not reception. In reading we can be purely in the receptive mode. I go from print to meaning when I read. In language production I must either speak or write. For computer language programs the student production mode is writing.

The second educational principle is that children learn language by modeling. They listen to adults speak and learn to speak. In many cases they listen to adults read and learn to read. They learn language naturally by modeling the language they

hear and see. Interactive language programs should provide the student with some form of language model.

The third educational principle is that the computer itself is a motivating factor. Most teachers will readily accept this point of view. The proliferation of home computing and computer gaming reinforces this point.

The fourth educational principle is that the student should not be a passive receiver of a computer program. As Dewey said a long time ago, children learn by doing.

In addition to observing these educational principles, one should follow certain guidelines when developing interactive language programs for the computer.

The computer should recognize the student; that is, call the student by name whenever possible.

```
10 INPUT "What is your name?"; N$
```

From then on the computer uses this name to interact with the student.

```
20 PRINT "Help me write a good sentence,"; N$;"."
```

This statement by the computer also immediately enlists the support of the student in a joint activity. The child is involved in helping the computer perform the task.

A second provision should allow the computer to learn from the student. In a sentence expansion exercise, the computer starts with a kernel sentence and asks the student to expand the sentence.

```
30 PRINT "Here is my sentence ⟶ The man laughed."
40 INPUT "Give me three words that describe the man.";A$,B$,C$
```

When the student responds with three adjectives such as tall, dark and handsome, the computer learns these three adjectives and responds with an expanded sentence.

```
50  PRINT "The";A$;" ";B$;" ";C$;"man laughed."
The tall, dark, handsome man laughed.
```

In order to keep a high level of interaction, the computer should respond positively to student efforts, regardless of the actual qualitative value of the student response. We know that negative feedback inhibits language production.

```
60 PRINT "Your sentence is more interesting than mine!"
```

Students should also have a chance to review their own language production and to rewrite if necessary. These revisions could be concerned with spelling, punctuation, usage or syntax.

```
70 INPUT "Look at this sentence. Do you want to make any corrections in spelling or punctuation, or change any of the words? (Y/N)";R$
```

A student's language production should also be reproduced in some permanent form. This allows the teacher to review it at a later time and allows the student to revise or expand on this language segment in the future. Two reproductive modes are available. If the computer has a printer attached, the computer can be instructed to reproduce the language segment. Since most classrooms will not have printing capacity, the student can be instructed to copy the language segment.

```
80 PRINT "Copy your sentence on your paper and show it to your
   teacher later."
```

Interactive programs should also be adaptable to change for continued use in the classroom without writing a completely new program. For example, the sentence writing program could be modified by starting with a different kernel sentence.

```
90 PRINT "My sentence is ⟶ People like."
```

The program can also ask the student different questions to expand the kernel sentence.

```
100 INPUT "How many people are they talking about?";R$
```

The results of such programs should be helpful in expanding students' language horizons. If all the students in the class completed the sentence expansion program and recorded their sentences, all sentences should be different. A class record of these sentences, along with a group discussion, should broaden student vocabularies, concepts and language capabilities. Completing such programs should help students understand how language is generated in written form. After such interactive programs, students should accept the fact that they can be writers!

References

Papert, S. *Mindstorms: Children, Computers, and Powerful Ideas.* New York: Basic Books, Inc., 1980.

Krutch, J. *Experiments in Artificial Intelligence for Small Computers.* Indianapolis: Howard W. Sams & Co. Inc., 1981.

These programs are written in BASIC for the TRS-80 III.

```
10  ' The purpose of this program is to motivate students to write
    poetry.
20  '
30  CLS : PRINT TAB(18) "W R I T I N G   P O E T R Y"
35  CLEAR 2000
40  PRINT TAB(20) "by GLORIA KUCHINSKAS" : PRINT TAB(23)
    "copyright, 1983"
50  PRINT STRING$(60,"*") :CLEAR 5000 :DATA 10,5,5,5
60  '
70  DATA mountain, lake, valley, man, deer, rock, fairy, child, star,
    mist
80  DATA soft, placid, limpid, implacable, fierce
90  DATA skillfully, swiftly, steadily, now, never
100 DATA be running, be standing, be sitting, be flowing,
    be shaded
110 '
120 READ N,J,A,V : DIM NN$(N), AJ$(J), AV$(A), VB$(V)
130 FOR I = 1 TO N : READ NN$(I) : NEXT I
140 FOR I = 1 TO J : READ AJ$(I) : NEXT I
150 FOR I = 1 TO A : READ AV$(I) : NEXT I
160 FOR I = 1 TO V : READ VB$(I) : NEXT I
170 '
190 PRINT TAB(10) "What is your first name ";: INPUT K$: PRINT
192 POKE 16409,0 :K$ = LEFT$(K$,INSTR(K$ + " "," ") - 1)
194 FOR X = 1 TO LEN(K$) :MID$(K$,X,1) = CHR$(ASC(MID$(K$,X,1))
    OR 32) :NEXT X
196 MID$(K$,1,1) = CHR$(ASC(K$) - 32)
200 PRINT TAB(5) "We are going to write a poem together,  "K$;"."
210 PRINT "I will write my poem first. Sometimes my poem doesn't
    make "
220 PRINT TAB(5) "sense. You may wish to ch- ,ge it. Here is my
    poem. . . ." :GOSUB 5[0
230 '
240 CLS: PRINT: PRINT TAB (14) "A  P O E M by the computer":
    PRINT
250 '           First Line : noun/in the/adjective/noun
260 D = RND(10) : PRINT TAB(14)NN$(D); "in the ";
270 G = RND(5) : PRINT AJ$(G);
280 J = RND(10) : IF J = D THEN 280   ELSE PRINT NN$(J)
290 '           Second Line : noun/adverb/adjective
300 E = RND(10) :IF E = D OR E = J THEN 300   ELSE PRINT
    TAB(14) NN$(E);
310 C = RND(5) : PRINT AV$(C);
320 H = RND(5) : IF G = H THEN 320    ELSE PRINT AJ$(H)
330 '           Third Line: when will the/noun/verb?
340 F = RND(10) : IF F = D OR F = C OR F = E OR F = J THEN 340
    ELSE 350
```

```
350 PRINT TAB(14) "when will the ";NN$(F);
360 C = RND(5) : PRINT VB$(C); "?" : PRINT : FOR I = 1 TO 1000 :
    NEXT I
370 '
380 PRINT TAB(5) "Does this poem make sense to you, ";K$;",
    (Y/N) ";
390 INPUT R$ :R$ = LEFT$(R$,1)
400 IF R$ = "Y" OR R$ = "y" THEN PRINT TAB(5) "You may copy
     my poem on paper if you want to.":GOSUB 550: GOTO 630:
     ELSE 410
410 IF R$ = "N" OR R$ = "n" THEN 420   ELSE 390
420 PRINT :PRINT TAB(12) "Retype the poem your way, ";K$;"."
430 PRINT TAB(3) "You may use punctuation and change the words
    if you like. "
440 PRINT "Type each line of poetry separately and press ENTER
    after each "
450 PRINT TAB(13) "line. Do not type the title." : LINEINPUT X$
460 LINEINPUT Y$ :LINEINPUT Z$ :FOR I = 1 TO 1000 :NEXT I
470 '
480 CLS :PRINT TAB(14) "A   P O E M   by ";K$ :PRINT
490 PRINT TAB(14) X$ : PRINT TAB(14) Y$ : PRINT TAB(14) Z$
500 PRINT : PRINT "Your poem is better than mine. But you and I
    sometimes "
510 PRINT "make mistakes, ";K$;". To be sure the spelling and "
520 PRINT "punctuation are correct, copy it down and show it to
    your "
530 PRINT TAB(22) "teacher later." :GOSUB 550 :GOTO 580
540 '
550 PRINT @910, "Press ENTER when you are ready.";
560 LINEINPUT W$
570 W$ = INKEY$ : IF W$ = " " THEN 560    ELSE RETURN
580 CLS:PRINT:PRINT TAB(10)"Do you want another poem ";K$;
    " (Y/N) ";
590 INPUT R$
600 IF LEFT$(R$,1) = "Y" OR LEFT$(R$,1) = "y" THEN CLS :
    GOTO 240
610 IF LEFT$(R$,1) = "N" OR LEFT$(R$,1) = "n" THEN END
    ELSE 590
620 '
```

```
10  '     The purpose of this program is to help students write richer
    sentences.
20  '
30  CLEAR 3000 :CLS :PRINT TAB(19) "SENTENCE EXPANSION"
40  PRINT TAB(18)"by GLORIA KUCHINSKAS"
50  PRINT TAB(20) "Copyright 1983" :PRINT STRING$(60,"*")
```

```
60  '
80  PRINT TAB(15) "What is your first name "; : INPUT N$
82  POKE 16409,0 :N$ = LEFT$(N$,INSTR(N$ + " "," ") - 1)
84  FOR X = 1 TO LEN(N$) :MID$(N$,X,1) = CHR$(ASC(MID$
    (N$,X,1)) OR 32) :NEXT X
86  MID$(N$,1,1) = CHR$(ASC(N$) - 32)
90  PRINT TAB(10) "Help me write a good sentence, ";N$;"." :FOR
    I = 1 TO 1000 :NEXT I
100 PRINT : PRINT TAB(10) "Here is my sentence: The man went."
110 PRINT TAB(32) STRING$(13,CHR$(131)) : FOR I = 1 TO 1000 :
    NEXT I
120 PRINT :PRINT TAB(5) "This is not a good sentence. It's not
    interesting."
130 PRINT "It doesn't tell us very much about the man and what he
    did."
140 PRINT TAB(10) "Help me make it more interesting, ";N$;"."
150 FOR I = 1 TO 1000 : NEXT I :GOSUB 500
160 '
170 CLS :PRINT TAB(5) "Give me three words that describe the
    man."
180 PRINT TAB(3) "Type your words one at a time under each
    other."
190 PRINT "Don't use any punctuation, ";N$;", but press ENTER
    each time."
200 LINEINPUT A$ : LINEINPUT B$ : LINEINPUT C$
210 PRINT "The ";A$;", ";B$;", ";C$;" man went." : FOR I = 1 TO
    1000 : NEXT I
220 '
230 PRINT :PRINT "Where did the man go?" :LINEINPUT D$ :FOR
    I = 1 TO 1000 :NEXT I
240 PRINT "The ";A$;", ";B$;", ";C$;" man went ";D$; "." :FOR I =
    1 TO 1000 :NEXT I
250 '
260 PRINT : PRINT "How did the man go ";D$;" ?" :LINEINPUT E$:
    FOR I = 1 TO 1000 :NEXT I
290 CLS :PRINT TAB(15) "Here is your sentence, ";N$;"."
300 PRINT "The ";A$;", ";B$;", ";C$;" man went ";D$;" ";E$;"."
310 PRINT STRING$(64,CHR$(131)) : FOR I = 1 TO 500 : NEXT I
320 PRINT :PRINT "Your sentence is much more interesting than
    mine. It tells "
330 PRINT "us much more about the man and what he did.
    However sometimes "
340 PRINT "you and I make mistakes in our work. To be sure we
    wrote our "
350 PRINT "sentence correctly, copy it down and show it to your
    teacher "
360 PRINT TAB(25) "later." :GOSUB 500
370 '
380 CLS :PRINT:PRINT: PRINT TAB(8) "You did such a good job. Do
    you want to work "
```

```
390 PRINT TAB(15) "with my sentence again (Y/N) ";
400 INPUT F$
410 IF LEFT$(F$,1) = "Y" OR LEFT$(F$,1) = "y" THEN 430
420 IF LEFT$(F$,1) = "N" OR LEFT$(F$,1) = "n" THEN END
    ELSE 400
430 PRINT : PRINT TAB(7) "Remember to use different answers this
    time, ";N$;"." :GOTO 100
490 '
500 PRINT : PRINT TAB(7) "Press ENTER when you are ready to go
    on."; : LINEINPUT Z$
510 Z$ = INKEY$ : IF Z$ = " " THEN 500    ELSE RETURN
```

```
10  '  This program helps students write paragraphs based on a
    main idea."
20  '
30  CLEAR 5000 :CLS :PRINT TAB(12) "W R I T I N G   A
    P A R A G R A P H"
40  PRINT TAB(20) "by GLORIA KUCHINSKAS" :PRINT TAB(23)
    "copyright 1983"
50  PRINT STRING$(60,131)
70  PRINT TAB(10) "What is your first name? ";: INPUT N$
72  POKE 16409,0 :N$ = LEFT$(N$,INSTR(N$ + " "," ") - 1)
74  FOR X = 1 TO LEN(N$) :MID$(N$,X,1) = CHR$(ASC(MID$
    (N$,X,1)) OR 32) :NEXT X
76  MID$(N$,1,1) = CHR$(ASC(N$) - 32)
80  '
90  PRINT :PRINT TAB(5) "You and I will write a paragraph
    together, ";N$;"."
100 PRINT TAB(5) "This is the first sentence of the paragraph . . ."
110 FOR I = 1 TO 1000 :NEXT I :GOSUB 770
120 PRINT @709, "The spaceman first saw Mars through his
    spaceport."; :GOSUB 750
130 FOR I = 1 TO 1000 :NEXT I :GOSUB 650  :FOR I = 1 TO 500 :
    NEXT I
140 '
150 CLS :GOSUB 770 :PRINT @197, "The spaceman first saw Mars
    through his spaceport.";
160 PRINT @389, "What will the paragraph be about, ";N$;",
    (a-c) ?"
170 PRINT "        a. outer space"
180 PRINT "        b. what Mars looked like
190 PRINT "        c. the journey home                    ";
200 INPUT R$ : X$ = LEFT$(R$,1)
210 IF X$ = "B" OR X$ = "b" THEN 260    ELSE 220
220 IF X$ = "A" OR X$ = "a" OR X$ = "C" OR X$ = "c" THEN 240
    ELSE 230
230 PRINT "Please type in a, b or c "; : GOTO 200
```

```
240 PRINT "Try again. The first sentence tells you what the
    paragraph"
250 PRINT "is about."                : GOTO 200
260 PRINT :PRINT "You are correct, ";N$;"!!! Yes, the paragraph
    will tell us "
270 PRINT TAB(5) "what the spaceman saw through the
    spaceports." :FOR I = 1 TO 1500 :NEXT I
280 '
290 CLS :L$ = "The spaceman first saw Mars through his
    spaceport." :GOSUB 770
300 PRINT @197, L$; :PRINT @389, "Type three sentences that tell
    what the spaceman saw."
310 PRINT "Type each sentence separately. Press ENTER after
    each sentence."
320 FOR I = 1 TO 3 :LINEINPUT A$(I) :NEXT I :FOR I = 1 TO 1000
    :NEXT I
330 '
340 CLS: PRINT L$: FOR I = 1 TO 3: PRINT A$(I): NEXT I:
    FOR I = 1 TO 1000: NEXT I
350 PRINT:PRINT "Look at your paragraph. Do all your sentences
    tell what the "
360 PRINT "spaceman saw when he looked out the spaceport? Do
    you need to "
370 PRINT "make any changes in your sentences? Do you need to
    make any "
380 PRINT TAB(7) "changes in spelling or punctuation?"
390 '
400 PRINT :PRINT TAB(3) "Do you need to change the paragraph,
    ";N$;", (Y/N) ";
410 INPUT R$ : X$ = LEFT$(R$,1)
420 IF X$ = "Y" OR X$ = "y" THEN 450   ELSE 430
430 IF X$ = "N" OR X$ = "n" THEN 530   ELSE 410
440 '
450 PRINT TAB(5) "How many lines do you need to type on ";N$;" "
451 INPUT L :IF L    8 THEN PRINT TAB(5) "You can only use 8
    lines.":FOR I = 1 TO 1000 :NEXT I :L = 8 :ELSE 460
460 CLS :PRINT L$ : FOR I = 1 TO L : PRINT A$(I) :NEXT I
470 PRINT "Retype the whole paragraph including the topic
    sentence."
480 PRINT "Type each line separately. Press ENTER after each
    sentence."
490 FOR I = 1 TO L :LINEINPUT S$(I) :NEXT I :FOR I = 1 TO 1000 :
    NEXT I
500 '
510 CLS :PRINT TAB(19) N$;"'s Paragraph" :PRINT
520 FOR I = 1 TO L : PRINT S$(I) : NEXT I : GOTO 550
530 CLS :PRINT TAB(22) N$;"'s Paragraph" : PRINT
540 PRINT L$ :FOR I = 1 TO 3 :PRINT A$(I) :NEXT I
550 PRINT : PRINT TAB(3) "I like your paragraph, ";N$;". However
    sometimes "
```

```
560 PRINT "you and I make mistakes when we write paragraphs.
    To be sure, "
570 PRINT "copy your paragraph on paper and show it to your
    teacher later." :GOSUB 750
580 '
630 PRINT :PRINT TAB(20) "GOOD-BY, ";N$;"!" :END
640 '
650 K$ = CHR$(179) : L$ = CHR$(191) : M$ = CHR$(140)
660 NN$ = CHR$(131) :P$ = CHR$(188) : Q$ = CHR$(143)
670 X$ = CHR$(176) :Y$ = X$ + P$ + L$ + L$ + L$ + P$ + X$
680 Z$ = Q$ + L$ + L$ + L$ + L$ + L$ + L$ + L$ + Q$
690 W$ = CHR$(77) + CHR$(65) + CHR$(82) + CHR$(83) + CHR$(195) +
    NN$ + Q$ + L$ + Q$ + NN$
700 CLS :PRINT @114, X$; :PRINT @175, Y$; :PRINT @238, Z$; :
    PRINT @297, W$;
710 CC$ = K$ + K$ + CHR$(187) + L$ + L$ + L$ + L$ + L$ + L$ + L$ +
    M$ + M$ :DD$ = CHR$(202)
720 FOR I = 453 TO 490 :PRINT @I, CC$ :FOR XX = 1 TO 10 :NEXT
    XX
730 PRINT @I, DD$ :NEXT :PRINT @I, CC$; :RETURN
740 '
750 PRINT @911, "Press ENTER when you are ready.";
760 LINEINPUT Z$ :Z$ = INKEY$ :IF Z$ = " " THEN 760    ELSE
    RETURN
770 PRINT :PRINT CHR$(151) + STRING$(62,131) + CHR$(171);
780 PRINT CHR$(149) + STRING$(62,128) + CHR$(170);
785 PRINT CHR$(149) + STRING$(62,128) + CHR$(170);
790 PRINT CHR$(181) + STRING$(62,176) + CHR$(186); :RETURN
```

Courseware Selection

Courseware Selection

by
Ann Lathrop
San Mateo County Office of Education

Educators have a crucial role to play in today's rapidly expanding field of instructional software. It is our responsibility to become skilled evaluators who look critically at courseware *before* we purchase it for use in our classrooms. We must demand excellence and reject that which is mediocre. We must be willing to write critical reviews in our professional journals. Finally, we must persuade those journals that have not yet added computer courseware to their reviews of instructional materials to do so.

The technology that can be used to set new standards of courseware excellence now exists. Creative designers and programmers are developing courseware that taps the interactive power of the computer and truly involves the student in the learning process. Merely placing workbook pages on the screen and asking occasional multiple-choice questions is no longer enough. We are well past the stage of being pleased simply because a program will load and run correctly. As we locate highly creative, interactive programs, we must publicize them to our associates for their own use and identify them for the larger educational community as a new standard for instructional computing.

Approximately 8,000 programs are currently being advertised for sale in the education market. Probably less than 10% of the programs fall into the category of good to excellent; some more conservative estimates place this figure at less than 5%. It is our challenge to select courseware to meet our students' needs from among the relatively few good programs now available.

THE EVALUATION PROCESS

Prior to beginning a critical review of courseware, it is helpful to select an evaluation instrument, guidelines or other standardized criteria. *The Guidelines for Evaluating Computerized Instructional Materials*, published by the National Council of Teachers of Mathematics, is one of the most carefully developed instruments. These guidelines are easy to read and have relatively simple forms. The guide is not geared just for mathematics, so teachers in any subject area will find it useful. The MicroSIFT *Evaluator's Guide for Microcomputer-Based Instructional Packages* is more complex, requiring careful study and a longer time to complete the forms. It is more appropriate for an in-depth analysis of a courseware package or for use by courseware developers or publishers. Each of these guides presents evaluation criteria and a thorough discussion of the evaluation process. A shorter evaluation form developed by The California Library Media Consortium for Classroom Evaluation of Microcomputer Courseware is designed as a training tool to aid teachers in identifying some of the important evaluative criteria.

The next step is to select and obtain courseware for review. The journals listed in this article publish critical reviews that can be helpful in making initial selections. These critical reviews are not to be confused with the publishers' announcements that are often reprinted from advertisements without any evaluation of the actual product. Critical reviews are best used to select programs for on-site evaluation. They should be considered to be a buying guide only as a last resort; previewing the courseware with the students who will be using it should be part of the selection process whenever possible.

Many courseware publishers and distributors now have a free 30-day on-approval policy, usually requiring an official purchase order. The primary advantage of ordering from such a source is that the courseware can be previewed in the classroom where student reactions will frequently modify an instructor's original opinion of the material. Courseware can also be previewed at conferences, software demonstrations, computer stores, district or regional centers, or at other schools. Some sales representatives will bring courseware to a school or district for preview. One innovative approach is the "software fair" or preview day to which

publishers, jobbers and educators from a large region are invited for the specific purpose of previewing a wide variety of courseware. All of these alternatives should be thoroughly explored before any courseware is ordered from a catalog description without the option of on-site preview before purchase.

After a courseware package has been obtained, there are three questions to be addressed before beginning a serious evaluation. In most cases a negative answer to any of the three may well eliminate the courseware from further consideration.

1. Does the program run on my equipment?
2. Does it meet a curriculum need at my school?
3. Does it represent a valid use of the computer?

These questions may appear too obvious to warrant discussion, but they are all too often ignored. Courseware is *not* transportable from one system to another and must match the exact configuration of equipment available at a specific site, including any required peripheral devices. Most courseware is selected to meet one or more stated curriculum objectives and should be evaluated in terms of those objectives. Even an outstanding program may be of little value if it does not fit into the curriculum. Finally, much of the courseware currently on the market appears to make only a trivial use of the computer. If the program merely replicates some task that is already being done well with a textbook, workbook or other traditional medium, its purchase would seem to be a waste of courseware funds.

EVALUATING THE PROGRAM

Several teachers, students and other staff members should become involved in the evaluation. Courseware is often used in more than one classroom and at several grade levels. Different teachers will emphasize different criteria. It is especially helpful for each teacher to use the package individually, preferably in the classroom, and then to discuss it critically with other reviewers before making a purchase decision. The evaluation steps outlined below are designed for one teacher, but they can be repeated by each person involved in the review.

1. **Be yourself.** Read the documentation, paying special attention to any stated or implied goals and objectives and to the instructions. When a management system is part of the courseware, try to assess how useful it might be and whether it will be easy to implement.

2. **Be a "good" student.** Go through the program in a positive manner. Follow instructions and try to do well. Ask the following questions:

Can I follow the instructions and understand what I am supposed to do?

Am I bored by the program or does it challenge me to perform well?

As a good student, have I learned anything or developed new concepts?

Was it fun? Would I want to run it again or use it with a friend?

3. **Be a "poor" student.** Make a great many errors. Get confused and try to return to the instructions for help. Miss the same problem/question several times in a row and see what happens. Then ask:

How do I feel about this program and about the computer?

How do I feel about myself? Did the program make me feel dumb or did it help me to feel successful?

Did the program help me when I made an error? Did it just say "try again" when I was already doing my best, or if I was guessing?

When I made an error, did the program branch to easier materials, present items more slowly, or explain the lesson in several different ways to help me?

Was there a beep or other noise that let the whole class know when I made an error?

Did I learn anything?

Would I ever want to use this program again?

4. **Be a "negative" student.** Press RETURN/ENTER unexpectedly. Ignore the instructions and press all of the wrong keys. Put in a number when the program asks for a letter. Be uncooperative. See how the program handles your antagonism:

Could I crash the program?

Did I get any insulting responses or did I get only a patient prompt that suggested what I should do?

Was it fun to fail? Did I get a more interesting graphic reward—the person was hanged, something exploded, the boat sank—when I gave the wrong answer?

Could I put a lot of crazy stuff on the screen or was the keyboard locked against unwanted responses?

Did I eventually get interested and become involved in the program almost in spite of myself?

These questions will help to identify truly creative and well designed courseware. Many programs deal fairly well with the good student unless they are, unfortunately, boring. It is in responding to student errors, intentional or otherwise, that a program designer has the opportunity to show imagination and to use the power of the computer to present material in new and more helpful ways.

Once you are familiar with the program, you will want to use it with your students, either individually, in small groups, or with the entire class. Students can be asked to complete some type of evaluation form or can be informally polled in discussions following the use of the program. Their reactions will provide additional insight into the potential value of the program being considered. Note especially whether most students complete the program without urging, want to repeat the program, or seem eager to share it with their friends. Then complete the evaluation form you have selected.

The final purchase/non-purchase decision should be based upon the opinions of the teachers involved, the reactions of students, and the relevance of the program to the curriculum. High standards must be established, and our final decision should reflect our determination to select only the very best from among the many programs available.

New courseware is appearing almost daily, and its quality is steadily improving. Any purchase should be deferred until there is enthusiastic agreement among the reviewers that is appropriate to the objectives of the school and truly represents an effective use of the computer. We control the market place by our decision to purchase or not to purchase a specific program and can encourage the development of creative and interactive programs by our refusal to purchase anything less.

SOFTWARE EVALUATION FORMS

The California Library Media Consortium for Classroom Evaluation of Microcomputer Courseware, "Checklist of Evaluative Criteria." Redwood City, California, 1983.

The Consortium developed this form with the goal of encouraging teachers to participate in the evaluation process by making the form as short and simple as possible. It is designed to be presented to teachers at a workshop, and there is no written documentation. The form may be reproduced and modified to meet local needs. (Available from Ann Lathrop, SMERC Library, San Mateo County Office of Education, 333 Main Street, Redwood City, CA 94063.)

Heck, William, Jerry Johnson, & Robert Kansky, *Guidelines for Evaluating Computerized Instructional Materials.* Reston, Virginia: National Council of Teachers of Mathematics, 1981.

This is a very usable booklet, especially for the novice. The evaluation criteria are well developed with clear explanations and examples, and the evaluation forms are simple and easy to complete. The guidelines are not math-specific and may be used effectively with courseware for any subject area. (Available from the National Council of Teachers of Mathematics, 1906 Association Drive, Reston, VA 22091.)

Isaacson, Dan, "Courseware Review and Rating Form," *The Computing Teacher* (December 1979/January 1980), pp. 12-13.

This two-page form may be reproduced without further permission for classroom use.

MicroSIFT, *The Evaluator's Guide for Microcomputer-Based Instructional Packages* (Fourth Printing). Eugene, Oregon: International Council for Computers in Education, 1983.

MicroSIFT has developed and thoroughly field-tested this comprehensive evaluation document designed to establish a model of excellence in courseware. The guidelines and forms are more complex than those from the National Council of Teachers of Mathematics, but the user who studies them carefully will learn a great deal about evaluation. Courseware developers will find this publication especially helpful. (Available from the International Council for Computers in Education, University of Oregon, 1787 Agate, Eugene, OR 97403-1923.)

REVIEW JOURNALS

Booklist
50 East Huron Street
Chicago, IL 60611

The Book Report
Box 14466
Columbus, OH 43213

California Library Media Consortium for Classroom Evaluation of Microcomputer Courseware
San Mateo County Office of Education
333 Main Street
Redwood City, CA 94063
$10 for each publication of 50 reviews:
Courseware Reviews: 1982
Courseware Reviews: 1983

Courseware Report Cart
150 West Carob Street
Compton, CA 90220

The Digest of Software Reviews: Education
1341 Bulldog Lane, Suite C
Fresno, CA 93710

EPIE Micro-Courseware Pro/Files
EPIE & Consumers' Union
Box 839
Water Mill, NY 11976

Media Review
Box 425
Ridgefield, CT 06877

MicroSIFT Reviews
Northwest Regional Educational Laboratory
300 S.W. Sixth Avenue
Portland, OR 97204
Free from participating network members

82

Peelings II
P.O. Box 188
Las Cruces, NM 88004

School Library Journal
Box 13706
Philadelphia, PA 19101

School Microware Reviews
Dresden Associates, Box 246
Dresden, ME 04342

S.E.C.T.O.R. Project
Exceptional Child Center
Utah State University
Logan, UT 84322
(published irregularly)

Software Review
520 Riverside Ave.
Westport, CT 06880

EDUCATION JOURNALS

Closing the Gap
Box 68
Henderson, MN 56044

Curriculum Review
517 S. Jefferson St.
Chicago, IL 60607

Instructor
7 Bank Street
Dansville, NY 14437

Journal of Learning Disabilities
11 East Adams St.
Chicago, IL 60603

Learning
Box 2580
Boulder, CO 80322

Media & Methods
1511 Walnut Street
Philadelphia, PA 19102

The Reading Teacher
Box 8139
Newark, DE 19711

Simulation & Games
275 South Beverly Dr.
Beverly Hills, CA 90212

EDUCATIONAL COMPUTING JOURNALS/MAGAZINES

CHIME
Clearinghouse of Information on
Microcomputers in Education
Oklahoma State University
108 Gundersen
Stillwater, OK 74078

Classroom Computer Learning
19 Davis Drive
Belmont, CA 94002

The Computing Teacher
University of Oregon
1787 Agate St.
Eugene, OR 97403-1923

CRLA (Computers, Reading and Language Arts)
Box 13039
Oakland, CA 94661

Educational Computer Magazine
Box 535
Cupertino, CA 95015

Educational Technology
140 Sylvan Avenue
Englewood Cliffs, NJ 07632

Electronic Education
1311 Executive Center Dr.
Suite 220
Tallahassee, FL 32301

Electronic Learning
902 Sylvan Avenue
Englewood Cliffs, NJ 07632

Microcomputers in Education
5 Chapel Hill Drive
Fairfield, CT 06432

Monitor
Association for Educational
Data Systems
1201 Sixteenth Street, N.W.
Washington, D.C. 20036

Teaching and Computers
902 Sylvan Avenue
Englewood Cliffs, NJ 07632

BIOGRAPHICAL NOTES

ALFRED BORK is a Professor of Physics and Director of the Educational Technology Center at the University of California at Irvine. Dr. Bork's work has included extensive experience developing computer-based learning materials. In addition, he has written over one hundred articles and several books, including *Learning with Computers* (Bedford, MA: Digital Equipment Corporation, 1981).

HUGH BURNS is with the Human Resources Laboratory at Lowry Air Force Base in Denver, Colorado. Major Burns is working on applied projects in artificial intelligence. He has written a number of articles and delivered several conference presentations on the use of computers in teaching composition. In 1980, Major Burns' doctoral study on computer-assisted invention was honored as the Outstanding Dissertation in the Humanities and Education at the University of Texas at Austin.

GLORIA KUCHINSKAS is an educational trainer/consultant specializing in microcomputer education in Boca Raton, Florida. Dr. Kuchinskas has edited several journals including the *Journal of Teacher Education* and *Computers, Reading and Language Arts.* In addition to developing software, Dr. Kuchinskas is in the process of publishing several books on using microcomputers in the reading and language arts classroom.

ANN LATHROP is the Library Coordinator for the San Mateo County Educational Resources Library and Microcomputer Center, Redwood City, California. Ms.Lathrop has taught at the elementary level, as well as serving as a school librarian. She is active in several professional organizations, including the California Media and Library Educators Association and the Computer-Using Educators. In addition, Ms. Lathrop writes a regular column for *Educational Computer* magazine.

JOSEPH LAWLOR is an Associate Member of the Professional Staff with SWRL Educational Research and Development, Los Alamitos, California. Mr. Lawlor's major projects at SWRL include reviewing sentence-combining research, designing sentence-combining instructional materials, and developing computer-based instructional programs. Mr. Lawlor has also written and co-authored a number of technical papers for SWRL and was the original editor for *Computers in Composition Instruction.*

STEPHEN MARCUS is the Assistant Director of the South Coast Writing Project at the University of California at Santa Barbara. Dr. Marcus has written several journal articles on computer-assisted writing instruction, in addition to delivering numerous presentations on this topic to professional organizations. Dr. Marcus also serves on the Santa Barbara County Task Force for Microcomputers in Education.

HELEN SCHWARTZ is an Associate Professor of English at Oakland University in Rochester, Michigan. In addition to developing software, Dr. Schwartz has set up and managed a computer writing lab, has acted as a consultant on computers and composition, and has started a lecture series and newsletter about computer use in research and instruction. Dr. Schwartz has written numerous articles and papers on computers and composition, and she is currently writing a book called *Communication and Computers* to be published by Holt Rinehart and Winston.

ROBERT SHOSTAK is a Professor of Education and Director of the Institute for Educational Technology at Florida International University in Miami. Dr. Shostak has published numerous articles on teaching English and has served as a consultant to several school districts. Dr. Shostak is also a contributing editor for the "Computers and the Teaching of English" column in *The Computing Teacher.*

EARL WOODRUFF is a graduate student at The Ontario Institute for Studies in Education, in Toronto, Ontario, Canada. As a member of the Writing Research Project conducted jointly by OISE and York University, Mr. Woodruff has assisted in developing and testing computer-based composition materials. Mr. Woodruff has also co-authored an article (with Carl Bereiter and Marlene Scardamalia) in the *Journal of Educational Technology Systems.*

WILLIAM WRESCH teaches English and computer science at the University of Wisconsin Center-Marinette, Marinette, Wisconsin. Dr. Wresch has expanded the number and kinds of computer science courses offered at the university, and he has directed a campus-wide CAI development project. Dr. Wresch is also an adjunct assistant professor at Northern Michigan University where he teaches graduate courses in computer-assisted instruction.

NOTES

ICCE Organization Members

Alaska Association for Computers in Education (AACE)
Contact: Sandra Burgess
Coordinator, The Northern Institute, 650 W. International Airport Rd., Anchorage, AK 99502

Alberta Association for Educational Data Systems
Contact: Keath Austen
838 Education Tower, University of Calgary
2500 University Dr. NW, Calgary
Alberta, Canada T2N 1N4

Arizona Computer Users in Education
Contact: Nancy Watson
2148 Saran Drive
Tempe, AZ 85281

Association for the Development of Computer-Based Instructional Systems (ADCIS)
Contact: Dr. Gordon Hayes, Executive Director
ADCIS - Miller Hall 409, Western Washington University, Bellingham, WA 98225

Computer Education Group of Victoria
Contact: Leon Gurr, Conference Secretary
P.O. Box 245, Nolline 3042
Melbourne, Victoria, Australia

Computer Education Group of Queensland
Contact: D.A. Carey
C.P.O. Box 1669
Brisbane, Queensland 4001
Australia

Computer Education Society of Ireland
Contact: Mary Devlin
85 Limetree Road East, Douglas, Cork, Ireland

Computer Educators of Idaho
Contact: Dennis Dempsey
P.O. Box 1068, Hailey, ID 83333

Computers, Learners, Users, Educators Association (CLUES)
Contact: Henry J. Petersen, Executive Director
50 Nellis Drive, Wayne, NJ 07470

Computer-Using Educators (California)
Contact: Carrie Fisher
Alameda County Office of Education
School Computer Unit
685 "A" St., Hayward, CA 94541

Computer-Using Educators of British Columbia (CUE/BC)
Contact: Barry Underwood
509 Alder St., Campbell River
B.C., Canada V9W 2N9

Computer-Using Educators of Kentucky (CUE/KY)
Contact: Janet Parker
EMCE Dept., School of Education
University of Louisville, Louisville, KY 40292

DIDACOM
Contact: Inno Bronkman
Asembeeck 96, 2182 RZ Hillegom
The Netherlands

Educational Computing Consortium of Ohio
Contact: Ellen Richman, Coordinator
4777 Fairhurst Road, Cleveland, OH 44124

Educational Computing Association of Western Australia
Contact: P.C. Tarrell, President
12 Lilac Place, Chamella 6062, Western Australia

Educational Computing Organization of Ontario (ECOO)
Contact: Robert E. Drake, President
252 Bloor Street West, Toronto
Ontario, Canada M5S 1V6

Educational Microcomputer Users Group of Central New York (EMUGCNY)
Contact: Jane McGrohan
Roxboro Middle School, Mattydale, NY 13211

Educators Interest Group of the San Diego Computer Society
Contact: Melvin L. Zeldches
P.O. Box 81537, San Diego, CA 92138

Florida Association of Science Teachers (FAST)
Contact: Richard Sweetser
1147 Scotten Rd., Jacksonville, FL 32205

Illinois Association for Educational Data Systems (ILAEDS)
Contact: Lyle B. Smith, President
Computer Science Department, Northern Illinois University, Box 128, De Kalb, IL 60115

Indiana Computer Educators
Contact: David A. Flowers
H. Wayne Community Schools, Adm. Cwt.
1230 So. Clinton St., Ft. Wayne, IN 46802

Manitoba Association for Educational Data Systems (MAN-AEDS)
Contact: Ed Boorsma, Secretary-Treasurer
1377 Wall Street, East Winnipeg, Manitoba
Canada R3E 2S5

Michigan Association for Computer Users in Learning (MACUL)
Contact: Larry R. Smith, Communications Sec.
33600 Van Born Road, Wayne, MI 48184

Minnesota Association for Educational Data Systems
Contact: Sue Talley
1925 W. County Rd. B2, St. Paul, MN 55113

Montana Council for Computers in Education
Contact: Leon Roland
1646 Bitterroot
Billings, MT 59105

National Institute for Microcomputer Based Learning (NIMBL)
Contact: Stanley Silverman, President
148 Plymouth Ave., Brightwaters, NY 11718

New Hampshire Association for Computer Education Statewide (NHACES)
Contact: Anne Knight, President
University of New Hampshire, Computer Services, Durham, NH 03824

New York State Association for Educational Data Systems
Contact: Don Boss, President
Ardsley High School, 800 Farm Road
Ardsley, NY 10502

New South Wales Computer Education Group
Contact: Toni Downes, Chairperson
P.O. Box 148, Broadway, N.S.W. 2007 Australia

Northwest Council for Computer Education (Oregon, Washington and Northern Idaho)
Contact: Howard Bailey
Computing Center, Eastern Oregon State College
La Grande, OR 97850

Oklahoma Educational Computer Users Program (OECUP)
Contact: Richard V. Andrew
Department of Mathematics, 601 Elm St., #423
The University of Oklahoma, Norman, OK 73019

Pennsylvania Learning Resources Association (PLRA)
Contact: George H. Zook, Director, IMS
Lancaster-Lebanon I.U. 13
P.O. Box 5026, Lancaster, PA 17603

Quebec Association of Computer Users in Education
Contact: Don Rosenbaum, Membership Director
P.O. Box 369, Ormstown, Quebec, J0S 1K0
Canada

Saskatchewan Association for Computers in Education
Contact: Duncan Campbell
Mt. Royal Collegiate, 2220 Rochdale Road
Saskatoon, Saskatchewan, Canada S7J 4A4

The Science Teachers' Association of Ontario
Contact: Warren Sirs
1032 Harkness Ave., Ottawa, Ontario
Canada K1V 6P2

Society of Data Educators (SDE)
Contact: Robert Behling
Bryant College, Smithfield, RI 02917

Society for Microcomputing in Life & Education (SMILE)
Contact: Hillel Weintraub, President
Doshisha International High School
Tatara, Tanabe-cho, Tsuzuki-gun
Kyoto-fu, Japan 610-03

Texas Computer Education Association
Contact: Vicki S. Smith
P.O. Box 2573, Austin, TX 78768

The Utah Council for Computers in Education
Contact: Dr. Larry C. Christensen
1295 North 1200 West, Mapleton, UT 84663

Wyoming Educational Computing Council
Contact: Robert L. Monssette, President
Laramie County School District #1, Admin. Bldg.
2810 House Ave., Cheyenne, WY 82001

Young People's LOGO Association (YPLA)
Contact: James H. Muller
1208 Hillsdale Drive, Richardson, TX 75081

ICCE PUBLICATIONS

The Computing Teacher journal

Computer Literacy Activities for Elementary & Middle School Students

Computer Metaphors: Approaches to Computer Literacy

Computer Technology for the Handicapped in Special Education & Rehabilitation: A Resource Guide

Computers in Composition Instruction

Evaluator's Guide for Microcomputer-Based Instructional Packages (Revised 11-83)

Introduction to Computers in Education for Elementary and Middle School Teachers

An Introduction to Computing: Content for a High School Course

Learning Disabled Students and Computers: A Teacher's Guide Book

Logo in the Classroom

Microcomputers in the Classroom—Dreams and Realities

Parent's Guide to Computers in Education

Precollege Computer Literacy: A Personal Computing Approach

School Administrator's Introduction to Instructional Use of Computers (Revised 4/83)

Teacher's Guide to Computers in the Elementary School

Topics: Computer Education for Colleges of Education